



TAMPEREEN TEKNILLINEN YLIOPISTO

TOMI TURTIAINEN
**AKTIIVI- JA RESERVIALUEOHJAUKSEN KEHITTÄMINEN VA-
RASTONHALLINTAJÄRJESTELMÄSSÄ**

Diplomityö

Tarkastaja: professori Hannu Koivisto
Tarkastaja ja aihe hyväksytty Teknisten
tieteiden tiedekuntaneuvoston kokouk-
sessa 5. kesäkuuta 2013

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO

Automaatiotekniikan koulutusohjelma

TURTIAINEN, TOMI: Aktiivi- ja reservialueohjauksen kehittäminen varastohallintajärjestelmässä

Diplomityö, 64 sivua

Joulukuu 2013

Pääaine: Automaatio- ja informaatioverkot

Tarkastaja: professori Hannu Koivisto

Avainsanat: varastohallintajärjestelmä, WMS, varaston ohjaus, keräily, aktiivialue, reservialue, täyttö

Yksi paljon käytetty tapa tehostaa keräilyä varastossa on käyttää aktiivi- ja reservialuejakoa. Aktiivialue on rajattu joukko varastopaikkoja, joista tuotteiden keräys on helppoa ja nopeaa. Työssä tutkitaan aktiivialueen ohjaamista varastohallintajärjestelmällä unohtamatta kuitenkaan muita varaston toimintaan liittyviä osakokonaisuuksia. Työn tavoitteena on kehittää Motus WMS -varastohallintajärjestelmän tekemää aktiivialueen ohjausta selvittämällä, voidaanko automaattisesti määrittää mitä nimikkeitä kannattaa aktiivialueelle varastoida ja miten paljon.

Työ jakaantuu kolmeen osaan: Kirjallisuustutkimusosassa käydään läpi varaston perusprosessit, niiden ohjaus ja simulointi sekä selvitetään mitä malleja aktiivialueen ohjaamiseen on kehitetty. Analysointitutkimusosassa tutkitaan miten aktiivialueen ohjaus on toteutettu Motus-varastohallintajärjestelmässä, analysoidaan mitä haasteita ja puutteita siinä on sekä esitetään teorian pohjalta, miten nimikkeiden ja määrien automaattinen määrittäminen voidaan toteuttaa Motuksessa. Simulaatiotutkimusosassa yhtä analysointitutkimusosassa esitettyä menetelmää verrataan Motuksen nykyiseen tapaan ohjata aktiivialuetta simuloimalla erään Motus-asiakkaan prosessia.

Tutkimus esittää kaksi menetelmää, joilla Motukseen voidaan toteuttaa automaattinen nimikkeiden ja määrien valinta kahdella erityyppisellä aktiivialueella. Toinen menetelmä soveltuu lavapohjaiselle aktiivialueelle, jossa täyttö pystytään tekemään lavasiirtona. Toinen menetelmä soveltuu pieniä nimikkeitä sisältävälle hyllypohjaiselle aktiivialueelle, jossa tuotteet varastoidaan suoraan hyllylle. Tutkimus osoittaa simuloinnin avulla, että ehdotettu menetelmä lavapohjaiselle aktiivialueelle kasvattaa keräilymääriä aktiivialueelta Motuksen nykyiseen malliin verrattuna. Tutkimus näyttää myös, että täyttöjen määrää voidaan vähentää varastoimalla nimikkeitä suoraan vastaanotosta aktiivialueelle reservialueen sijaan.

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY

Automation engineering

TURTIAINEN, TOMI: Improving forward reserve control in a warehouse management system

Master's thesis, 64 pages

December 2013

Major: Automation and information networks

Examiner: professor Hannu Koivisto

Keywords: warehouse management system, warehousing, forward pick area, forward reserve control, replenishment

A well-known method to make picking more efficient in a warehouse is to use both a separate forward and reserve area. The forward area refers to that of a storage area within a warehouse, from where accessing items is fast and convenient, and therefore picking items becomes more efficient than that from a reserve area. This thesis examines how the forward area can be controlled using a warehouse management system (WMS) without forgetting other warehouse processes. The main goal is to improve the forward area control in a Motus WMS by examining if the Stock-Keeping Units (SKUs) and their quantities that are assigned to forward area could be automatically determined.

The thesis is divided into three parts: In the literature study part, basic warehouse processes and how these are controlled and simulated are explored. Earlier research on forward reserve control is explored. During the analysis part of the thesis, the forward reserve control in Motus WMS is analyzed on how it is currently being done and what weaknesses it has. Based on theory and analysis, new methods to improve the forward reserve control are developed and explained. In the simulation study part, one of the methods stated in the previous part is then compared to Motus WMS' current method to control the forward area using a model and data from one of Motus WMS' customers.

The study states two methods for controlling two different types of forward area that can be used in Motus WMS. While one method is suitable for a forward area that consists of pallet racks where replenishments are a unit-load process, the other method is suitable for a forward area that uses shelving for smaller products. When compared to the current method Motus WMS uses, based on the simulations, there is an indication that a higher amount of picking is achieved in the forward area via the stated method for pallet rack forward area. The study also indicates that the amount of replenishments can be decreased by using so called Dock-To-Forward technique, which stores products directly to forward area from reception.

ALKUSANAT

Diplomityön tekeminen on ollut pitkä prosessi aiheen valinnasta käytännön toteutuksen kautta itse kirjoittamiseen. Vaikkakin olen vastannut itse kaikista diplomityöprosessin vaiheista, ovat saamani tuki ja neuvot eri henkilöiltä olleet ensiarvoisen tärkeitä. Aihe valikoitui selaamalla laajasti varastonohjaukseen liittyvää kirjallisuutta sekä varastonhallintajärjestelmässä havaittujen kehityskohteiden kautta.

Työn toteutuksen aikana saaduista kommentteista ja kannustuksesta haluan kiittää esimiestäni Business Unit Manager Anssi Tikkaa. Haluan myös kiittää Senior Architect Jani Jokirannalle teoriakatsaukseen ja työn suosituksiin liittyvistä huomioista ja ideoista.

Kiitokset ansaitsee myös professori Hannu Koivisto Tampereen teknilliseltä yliopistolta. Hän auttoi työn aiheen rajauksessa sekä tutkimusongelman muotoilussa. Häneltä työstä saadut kommentit olivat myös arvokkaita työn lopputuloksen kannalta.

18.11.2013

Tomi Turtiainen

SISÄLLYS

1. Johdanto	1
2. Varastotoiminnot	3
2.1 Vastaanotto	4
2.2 Varastointi	4
2.2.1 Täytöt	4
2.3 Keräily	5
2.4 Cross-docking	5
2.5 Pakkaus	5
2.6 Lähetys	6
3. Varastonhallintajärjestelmä	7
3.1 Motus WMS -varastonhallintajärjestelmä	7
4. Varastonohjaus	9
4.1 Varastointipolitiikka	9
4.2 Keruustrategia	10
4.2.1 Niputus	11
4.2.2 Reititys	12
4.2.3 Keruualueet	13
5. Varastoprosessien mallinnus ja simulointi	15
5.1 Differentiaaliyhtälömalli	15
5.2 Tapahtumapohjainen malli	16
5.3 Agenttipohjainen mallinnus	17
5.4 Varastoprosessien simulointi	18
6. Aktiivi- ja reservialue	21
6.1 Aktiivialueiden luokittelu	22
6.1.1 Hyllypohjaisen aktiivialueen ohjaus	22
6.1.2 Lavapohjaisen aktiivialueen ohjaus	27
6.2 Täytöt	30
7. Aktiivialueohjaus Motuksessa	34
7.1 Varastointisäännöt	35
7.2 Keräyssäännöt	35
7.3 Täyttösäännöt	36
7.4 Aktiivialueen mallintaminen	37

7.4.1	Varastointi- ja täyttösääntöjen määrittäminen Motuksessa	37
7.5	Varastointi- ja täyttösääntöjen rajoitteet	39
8.	Täyttösääntöjen automaattinen määrittäminen	41
8.1	Mallien oletukset	41
8.2	Keruutehtävien ja täyttöjen suoritusajan estimointi	43
8.3	Kysynnän estimoiminen historiatiedoista	44
8.3.1	Kysynnän estimoiminen nestemallia varten	45
8.3.2	Kysynnän estimoiminen lavapohjaista mallia varten	45
8.4	Lavapohjaisen mallin muut parametrit	45
8.5	Mallien tulosten muuntaminen täyttösäännöiksi	46
8.6	Saldon huomioiminen	47
8.7	Varastointisääntöjen automaattinen määrittäminen	47
8.8	Menetelmä hyllypohjaiselle aktiivialueelle	47
8.9	Menetelmä lavapohjaiselle aktiivialueelle	48
9.	Automaattisten täyttösääntöjen vertailu simuloinnilla	50
9.1	Simuloitava prosessi	50
9.2	Simuloinnin toteutus	52
9.3	Simuloinnin tulokset	55
10.	Johtopäätökset	59
	Lähteet	61

LYHENTEET, TERMIT JA NIIDEN MÄÄRITELMÄT

Motus	Leanware Oy:n kehittämä varastohallintajärjestelmä.
Ottokerta	Yksi nimikkeen keräilytapahtuma, jossa nimikettä kerätään varastosta.
Täyttö	Tuotteiden siirtäminen varaston sisällä alueelta toiselle. Yleensä täyttö tehdään reservialueelta aktiivialueelle.
Täyttöaste	Suhdeluku, joka kertoo montako prosenttia tietyn varastoalueen varastopaikoista on varattu.
Varastoalue	Loogisesti ryhmitelty joukko varastopaikkoja, esimerkiksi reservipaikat.
WMS	Warehouse Management System, eli varastohallintajärjestelmä. Tietojärjestelmä, joka ohjaa varaston operatiivista toimintaa.
Yksikkö	Säilytysalusta, jonka päälle tai sisään nimikkeitä voidaan säilöä. Esimerkiksi kuormalava, muovilaatikko tai rullakko.

1. JOHDANTO

Maailmankaupan nopean kasvun seurauksena toimitusketjut ovat pidentyneet ja pirstaloituneet, minkä vuoksi varastojen rooli niiden osana on korostunut. [1] Varastojen avulla muun muassa vähennetään kausi-, kuljetus- ja tuotantovaihteluiden vaikutuksia materiaalivirtaan, yhdistetään useammalta toimittajalta tulevia materiaalivirtoja samaan toimitukseen sekä tarjotaan lisäarvopalveluja, kuten esiasennusta ja kokoonpanoa. [2] Varastot ovatkin nykyään kriittinen osa toimitusketjua ja niiden tehokkaalla ja lisäarvoatuottavalla toiminnalla voidaan saavuttaa kilpailuetua. [3]

Varastojen sisäpuolella tapahtuvaa materiaalinkäsittelyä kutsutaan sisälogistii-kaksi. Sisälogistiikassa selvästi kallein kustannuserä on keräily. Sen osuus kaikista operatiivisista työkustannuksista on viimeisimmän tutkimuksen mukaan lähes 40 %. [1] Keräilyn tehostamiseksi varastopaikat on monissa varastoissa jaettu aktiivi- ja reservialueeseen. Kerääminen aktiivialueelta on helppoa ja nopeaa, mutta tilaa on rajallisesti. Reservialueelta kerääminen taas on hitaampaa, mutta tilankäyttö tehokkaampaa. Aktiivialuetta joudutaan myös täyttämään reservialueelta kun tuotteet loppuvat aktiivialueelta, mikä aiheuttaa itsessään turhaa lisätyötä.

Varaston tehokkaaseen ohjaamiseen tarvitaan varastohallintajärjestelmää. Varastohallintajärjestelmä ohjaa varaston toimintaa ja vastaa muun muassa nimikkeiden varastopaikan valinnasta sekä nimikkeiden siirtämisestä reservialueelta aktiivialueelle. [3] Eräs varastohallintajärjestelmä on Leanware Oy:n kehittämä Motus WMS.

Tämän diplomityön päätavoite on kehittää Motus WMS varastohallintajärjestelmän aktiivialueen käytön ohjaamista. Tutkimuksen keskeisin kysymys on ”Miten Motus WMS:in aktiivialueen käytön ohjaamista voidaan tehostaa?”. Tyypilliset päätökset aktiivi- ja reservialuetta käytettäessä ovat miten nimikkeet jaotellaan niiden välillä, sekä miten paljon valittua nimikettä kannattaa sijoittaa aktiivialueelle. Nämä kysymykset varastohallintajärjestelmän näkökulmasta muodostavat tutkimuksen osaongelman: ”Pystytäänkö aktiivialueelle varastoitavat nimikkeet ja määrät laskemaan automaattisesti?”. Toinen keskeinen ongelma aktiivi- ja reservialueen käyttöön liittyy täyttöjen suorittamiseen, joka muodostaa tutkimuksen toisen osaongelman: ”Milloin ja miten paljon nimikettä kannattaa siirtää reservialueelta aktiivialueelle?”.

Kaikista varastotoiminnoista on olemassa enemmän tai vähemmän aiempaa tutkimusaineistoa, joskin enemmistö tutkimuksista käsittelee keräilyä sen suuren kustannusosuuden takia. Useimmat tutkimukset keskittyvät yhden tietyn varaston

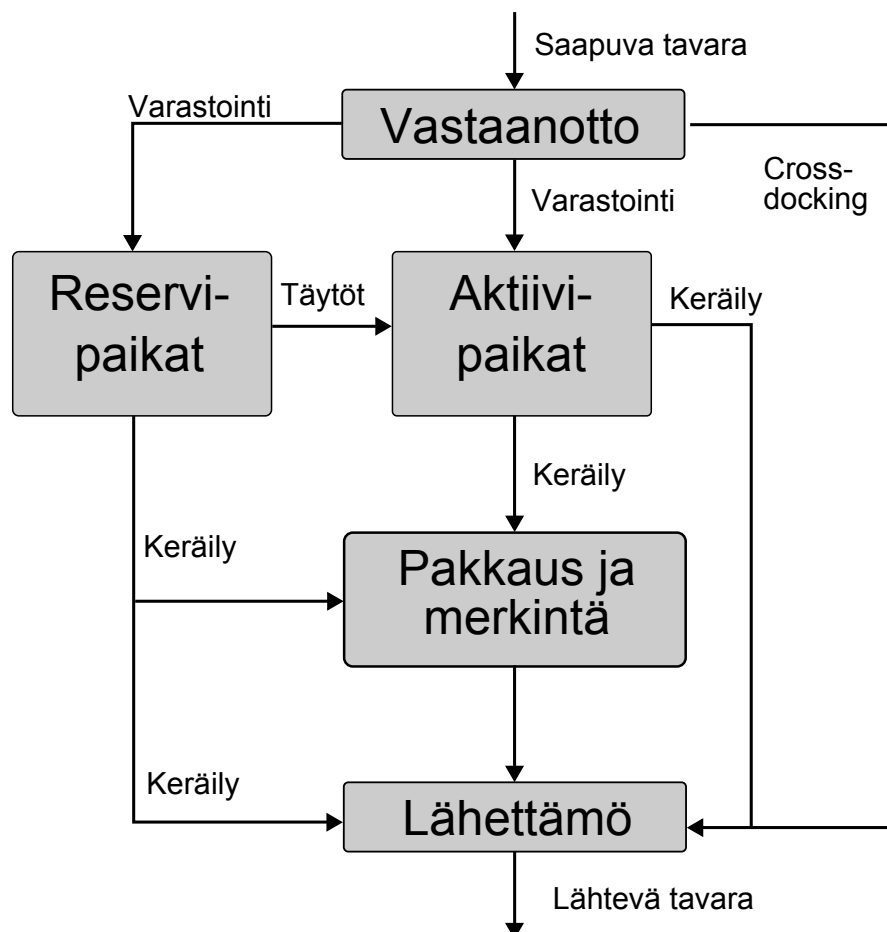
osaprosessin analysoimiseen. Varasto erilaisine toimintoineen on kuitenkin hyvin monimutkainen kokonaisuus, jossa eri toiminnot vaikuttavat suorasti ja epäsuorasti toisiinsa. Tämän tutkimuksen tavoitteena on tarkastella aktiivi-reservialueohjausta varastohallintajärjestelmän näkökulmasta kokonaisuutta kuitenkin unohtamatta. Tarkastelu pohjautuu puhtaasti varaston sisäiseen, operatiiviseen toimintaan, eikä esimerkiksi ulkoisia kustannustekijöitä, kuten ostokuluja tai varastoon sitoutunutta pääomaa, oteta tässä työssä huomioon.

Työ koostuu kolmesta vaiheesta. Ensimmäisessä vaiheessa tehdään kirjallisuuskatsaus olemassa olevaan tutkimusaineistoon aktiivialueohjaukseen liittyen. Toisessa vaiheessa analysoidaan nykyinen Motuksen suorittama aktiivialueohjaus ja esitetään ensimmäisen vaiheen tuloksia hyödyntäen uudet ohjausta parantavat menetelmät. Viimeisessä vaiheessa nykyistä ja parannettua aktiivialueohjausta verrataan simuloimalla erään Motus asiakkaan prosessia.

Työ on jäsennetty seuraavasti. Kappaleessa 2 on kuvattu yleisellä tasolla varaston perusprosessit ja kappaleessa 3 varastohallintajärjestelmien tarkoitus ja hyödyt. Kappaleessa 4 on esitetty miten varaston ohjausta tehdään ja mitä problematiikkaa siiheen liittyy. Kappaleessa 5 on esitetty simuloinnin hyödyt mallinnuksessa ja miten varastoprosesseja simuloidaan. Kappaleessa 6 on tehty kirjallisuuskatsaus aiempaan aktiivialueohjaukseen liittyvään tutkimukseen ja kappaleessa 7 on analysoitu, miten aktiivialueen ohjaus on nykyään toteutettu Motuksessa ja mitä kehitettävää siinä on. Kappaleessa 8 on esitetty analyysin pohjalta uudet menetelmät aktiivialueohjauksen tehostamiseksi, joista erästä on vertailtu nykyiseen ohjaukseen simuloimalla kappaleessa 9. Lopulta työn johtopäätökset on koottu kappaleessa 10.

2. VARASTOTOIMINNOT

Vaikka varastot prosesseineen voivat erota hyvinkin paljon toisistaan, ovat niiden materiaalivirrat yleensä hyvin samantyyppisiä, kuten kuvassa 2.1 on esitetty. Saapuva tavara otetaan vastaan ja varastoidaan keräämistä varten. Asiakkaan tilausten perusteella nimikkeet kerätään ja lähetetään asiakkaille. Tästä materiaalivirrasta voidaan tunnistaa neljä päätoimintoa, jotka löytyvät lähestulkoon jokaisesta varastosta. Ne ovat saapuvan tavaran vastaanotto, vastaanotetun tavaran varastointi, tilattujen tuotteiden keräily sekä lähetys. Näiden lisäksi tilauksia usein myös pakataan, merkitään, yhdistellään ja lajitellaan ennen lähettämistä. [4, 5] Tässä kappaleessa tutustutaan tarkemmin näihin toimintoihin.



Kuva 2.1: Varaston materiaalivirrat

2.1 Vastaanotto

Vastaanotettava tavara saapuu varastolle kuormina. Saapuva kuorma puretaan ja rekisteröidään saldoille, jotta tuotteiden tiedetään olevan saatavilla asiakastilauksia varten. Saapuneet kuormat myös tarkistetaan mahdollisten vaurioiden ja virheellisten määrien tai tuotteiden varalta ja poikkeamat kirjataan ylös. [4] Saapuneille tuotteille voidaan myös suorittaa laaduntarkastuksia.

Vastaanoton yhteydessä saatetaan myös suorittaa yksiköintiä, mikäli saapunutta tavaraa ei voida varastoida samalla alustalla, kuin millä se saapui. Tällöin tuotteet siirretään uusiin säilytysalustoihin eli yksiköihin, esimerkiksi lavalta muovilaatikoihin. [6] Vastaanoton osuus varastokustannuksista tyypillisessä jakelukeskuksessa Suomessa on noin 9 prosenttia. [1]

2.2 Varastointi

Varastointi koostuu varastopaikan valinnasta, johon varastoitavat nimikkeet varastoidaan, sekä varsinaisesta nimikkeiden siirtämisestä valittuihin varastopaikkoihin. Varastopaikan valinta on toimenpide, jossa valitaan varastointisääntöjen perusteella varastopaikka, mihin tarkastelun kohteena olevat nimikkeet varastoidaan. [4]

Sijoituslaskenta voidaan jakaa kahteen osaan; varastonosan valintaa ja varsinaisen varastopaikan valintaan valitun varastonosan sisällä. [5] Varastonosan valinta voi olla esimerkiksi valinta automaattikarusellin ja lavavaraston välillä. Tällöin varastopaikan valinta olisi karusellin hyllyn ja lokeron valinta tai vastaavasti lavavaraston hyllypaikan valinta. Varastonosan valinta liittyy suoraan vastaanotossa tehtävään yksiköintiin, sillä eri varastonosat vaativat usein omanlaiset pohja-alustat (esim. lavavarastossa lava, automaattikarusellissa muovilaatikko). Varastoinnin osuus varastokustannuksista Suomessa on noin 8 prosenttia. [1]

2.2.1 Täytöt

Osaksi varastointia voidaan lukea myös täytöt. [5] Täytöillä tarkoitetaan nimikkeiden siirtämistä yhdestä varastopaikasta toiseen varaston sisällä. [6] Täyttöjen avulla pysytään ylläpitämään tietyllä varastoalueella aina tietty määrä nimikkeitä. Esimerkiksi keräily on yleensä nopeampaa lavavaraston lattiapaikoilta kuin yläorsilta, minkä vuoksi nimikkeitä halutaan pitää tietty määrä lattiapaikoilla.

2.3 Keräily

Keräily on toiminto, jossa varastosta noudetaan oikea määrä oikeita tuotteita tilausten perusteella. Tilaus määrittää yhdelle asiakkaalle¹ toimitettavat nimikkeet ja se koostuu tilausriveistä. Yksi tilausrivi määrittää nimikkeen ja määrän, paljonko sitä toimitetaan. [4]

Keruujärjestelmät voidaan luokitella sen mukaan, miten paljon automaatiota niissä hyödynnetään. Yksinkertaisin ja yleisin tapa tehdä keruuta on manuaalikeräys, jossa kerääjä kulkee varastopaikasta toiseen keräten tuotteet (engl. picker-to-parts). Automaattivarastoissa hyödynnetään vertikaalikaruselleja sekä hyllystöhisseejä, jotka tuovat nimikkeet nk. keruuasemaan, josta keräys suoritetaan (engl. parts-to-picker). Täysin automatisoiduissa keruujärjestelmissä keräys tapahtuu automaattisesti esimerkiksi robotin toimesta. Täysin automatisoidut keruujärjestelmät ovat ainakin toistaiseksi vielä harvinaisia. [7]

Keräily on operatiivisilta kustannuksiltaan kallein toiminto varastossa. Sen osuus kaikista varastokustannuksista yhdessä sisäisten siirtojen kanssa on Suomessa keskimäärin noin 38 prosenttia. [1] Keräilytoiminto voidaan jakaa kulkemiseen, etsimiseen, poimimiseen ja muihin toimintoihin. Näistä selvästi eniten aikaa vie kulkeminen, noin 55 prosenttia. [4] Tästä syystä varaston toiminnan tehostamisessa keskitytään useimmiten juuri keräilyn tehostamiseen, missä keskitytään yleensä tuottamattoman kulkuajan ja -matkan pienentämiseen.

2.4 Cross-docking

Cross-docking on tekniikka, jolla pyritään vähentämään nimikkeiden käsittelykustannuksia pääsemällä kokonaan eroon keräilystä ja varastoinnista. Vaatimuksena on, että nimikkeille on olemassa tilaaja, ennen kuin nimikkeet otetaan vastaan. Tällöin ne voidaan siirtää vastaanotosta suoraan lähettämöön ilman varastointia ja keräilyä.

2.5 Pakkaus

Pakkauksen tarkoituksena on paitsi pakata tuotteet kuljetusta varten myös yhdistää eri materiaalivirrat. Samalle asiakkaalle lähtevät tuotteet pyritään pakkaamaan samaan yksikköön kuljetuskustannusten minimoimiseksi. Yksikkö on pohja-alusta, jolla varastoidaan nimikkeitä, kuten esimerkiksi kuormalava tai pahvilaatikko. Samassa yhteydessä kerätyt tilaukset myös tarkistetaan virheiden minimoimiseksi. [4]

¹Asiakkaalla tarkoitetaan tässä osapuolta, jolle nimikkeet toimitetaan. Esimerkiksi jakelukeskuksessa asiakas on tuotteet tilannut vastaanottaja, kun taas materiaalivarastossa asiakas voi olla esimerkiksi tuotantosolu.

2.6 Lähetys

Lähtetäessä keräyksessä ja pakkauksessa muodostuneita yksiköitä voidaan vielä lajitella ja yhdistellä. Yhdistelyä tehdään kuljetuskustannusten pienentämiseksi ja lajittelua esimerkiksi kun yksiköt pitää lastata päinvastaisessa järjestyksessä kuin missä ne toimitetaan. Lopulta tilaukset lastataan rekkaan tai muuhun kuljetusvälineeseen ja kuitataan luovutetuksi varastolta. [4]

3. VARASTONHALLINTAJÄRJESTELMÄ

Varastonhallintajärjestelmä (engl. warehouse management system, WMS) ohjaa kaikkia varaston toimintoja, joista on tarkempi kuvaus edellisessä kappaleessa. Se kokoaa, varastoi ja raportoi reaaliaikaista tietoa, jota tarvitaan materiaa livirtojen tehokkaaseen ohjaamiseen ja hallintaan. [3]

WMS tuntee kaikki varaston tuotteet sekä niiden sijainnin, pakkauskoon ja fyysiset mitat. Lisäksi se tuntee varaston varastopaikat sijainteineen ja fyysisine mittoineen. Näitä tietoja hyödyntämällä se ohjaa ja hallitsee varaston materiaa livirtoja sekä resursseja. Varastonhallintajärjestelmän avulla pystytään vähentämään virheitä, parantamaan tuottavuutta ja tarjoamaan uusia lisäarvoa tuottavia palveluita. [3, 4]

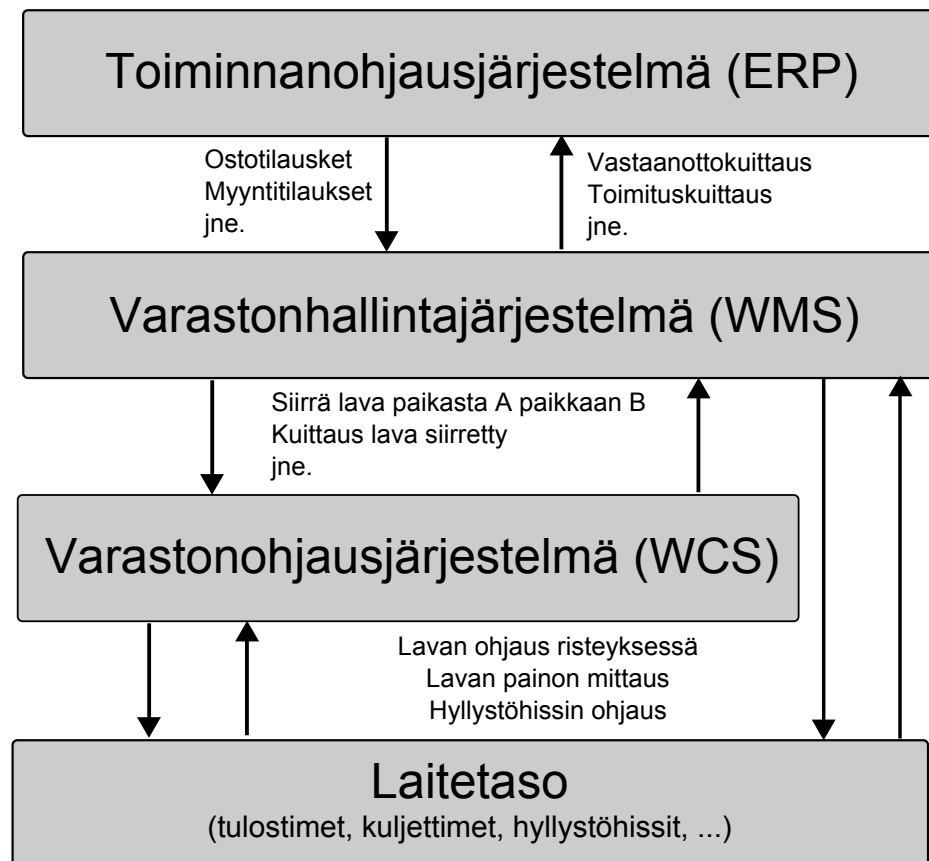
Kuvassa 3.1 on kuvattu miten varastonhallintajärjestelmä sijoittuu suhteessa muihin järjestelmiin. Varastotoiminnot palvelevat aina jotakin liiketoimintaprosessia, joita ohjataan yhä useammin toiminnohjausjärjestelmällä (engl. enterprise resource planning system, ERP). [3] Esimerkiksi tavaran vastaanottaminen saldoille käynnistää laskun maksamisen toimittajalle ja tavaran lähettäminen käynnistää laskun lähettämisen vastaanottajalle. Tästä syystä WMS:n täytyy pystyä kommunikoimaan liiketoimintaprosesseista vastaavan järjestelmän kanssa.

Vastaavasti WMS:in täytyy kyetä kommunikoimaan teknisten laitteiden ja järjestelmien kanssa, jotta se pystyy ohjaamaan tuotteiden käsittelyä ja liikuttamista varaston sisällä. Tällaisia järjestelmiä ovat esimerkiksi tulostimet, hyllystöhissit, kuljettimet, RFID-järjestelmät tai kaikkea varastoautomaatiota ohjaavat varastonohjausjärjestelmät (engl. warehouse control system, WCS). [3]

3.1 Motus WMS -varastonhallintajärjestelmä

Motus WMS on Leanware Oy:n kehittämä varastonhallintajärjestelmä. Leanware Oy on Tampereella sijaitseva vuonna 1999 perustettu logistiikan ja teollisuuden ohjelmistotalo. Leanware Oy:llä on kirjoitushetkellä yli 70 työntekijää ja se koostuu Logistics-, Industry- ja Informatics-liiketoimintayksiköistä. Logistics-yksikkö kehittää ja ylläpitää Motus WMS -järjestelmää, joka on käytössä kymmenkunnalla asiakkaalla, joihin lukeutuu useita suuria Suomen markkinoilla toimivia yrityksiä.

Motus WMS on kehittynyt, C#-ohjelmointikielellä ja Microsoftin .NET-ohjelmistokomponenttikirjastolla toteutettu varastonhallintajärjestelmä. Motus on mahdollista integroida niin ERP-järjestelmiin kuin varastoautomaatioonkin, minkä vuoksi se so-



Kuva 3.1: Varastohallintajärjestelmän suhde muihin tietojärjestelmiin

veltuu sekä automaatti- että manuaalivarastojen ohjaamiseen. Automaattivarastoissa Motus voi ohjata varastoautomaatiota joko suoraan tai varastonohjausjärjestelmän välityksellä.

Motus WMS kykenee ohjaamaan kaikkia kappaleessa 2 esitettyjä varaston perusprosesseja. Näiden lisäksi se kykenee myös vaativampien prosessien, kuten kokoonpanon ja esiasennuksen, ohjaukseen. Ohjaamisen lisäksi Motuksessa on kehittyneitä toiminnallisuuksia esimerkiksi jäljitettävyyteen, raportointiin, palautusten hallintaan ja EDI-sanomien lähetykseen. [8]

Valittaessa varastopaikkaa valitaan satunnaisesti jokin vapaa varastopaikka, jolloin jokaisella vapaalla varastopaikalla on yhtä suuri todennäköisyys tulla valituksi. [7]

Käytettäessä lähin vapaa -varastointipolitiikkaa valitaan varastopaikaksi kulkumatkaltaan lähin vapaana oleva varastopaikka. Vertailupisteenä käytetään varastointin aloituspistettä, joka yleensä on automaattivarastoissa kraanan syöttöasema ja manuaalivarastoissa vastaanottoalue. [7]

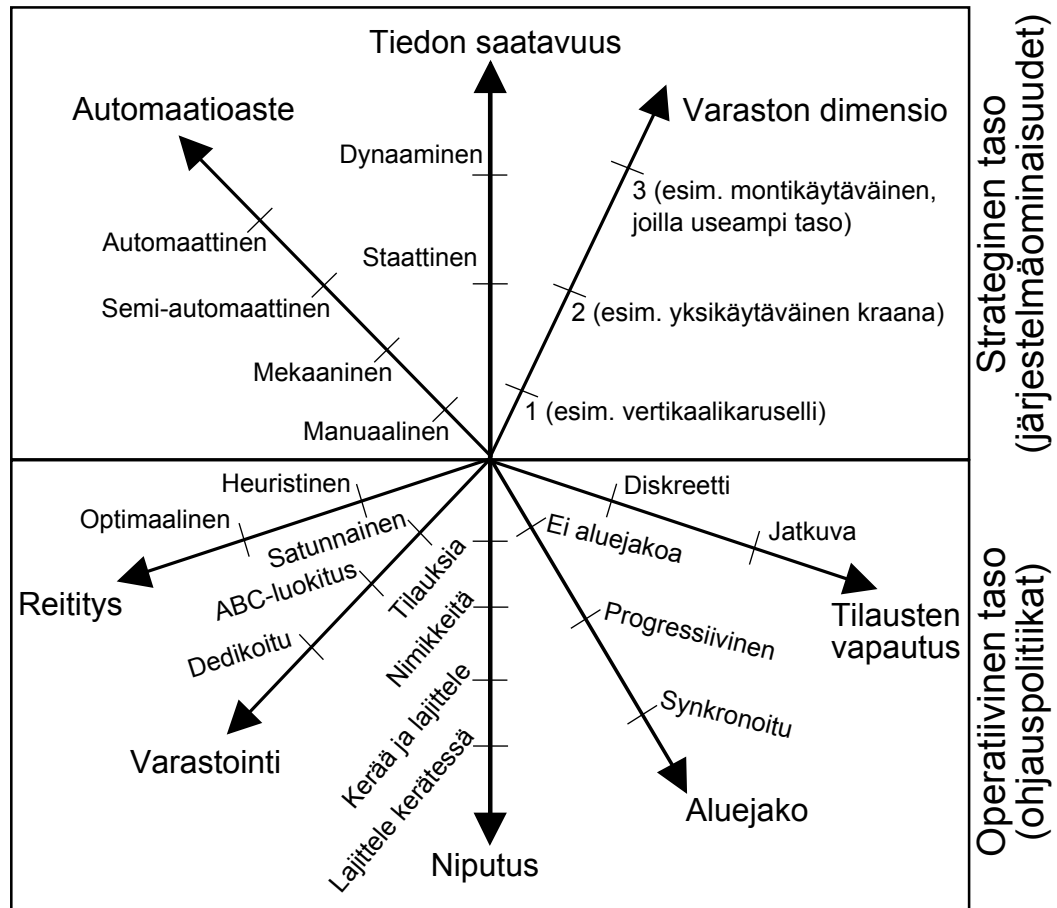
Dedikoitua varastointipolitiikkaa käytettäessä jokainen varastopaikka varataan tietylle nimikkeelle, jolloin siihen voidaan sijoittaa vain valittua nimikettä. Suositummat nimikkeet sijoitetaan keräilyn kannalta suotuisammille paikoille, kuten lähemmäs syöttöasemaa tai lähettämöä. Nimikkeiden suosio voidaan määrittää esimerkiksi kuutiota tilauksessa -suhdeluvun (engl. cube-per-order index, COI), keräilymäärien (engl. volume-based storage, VBS) tai pelkän menekin perusteella. [7]

Luokkapohjainen varastointipolitiikka (engl. class-based storage, CBS), jota usein kutsutaan myös ABC-luokitteluksi, pyrkii yhdistämään dedikoidun ja satunnaisen politiikan parhaat puolet. Nimikkeet jaetaan luokkiin jonkin muuttujan, yleensä menekin, mukaan ja jokaista luokkaa vastaa tietty joukko varastopaikkoja. Luokan sisällä nimikkeen varastopaikka valitaan satunnaisesti. [7] Petersen ym. [10] suosittelevat luokkien määräksi manuaalivarastossa 2-4, kun taas van den Bergin ym. [11] tutkimuksessa kuusi luokkaa antoi paremman tuloksen automaattivarastossa. Muita päätöksiä CBS-politiikkaan käyttöön liittyen ovat kuinka suuri osa nimikkeistä sijoitetaan mihinkin luokkaa ja miten varastopaikat jaetaan luokkiin.

Tilankäytön kannalta tehokkain varastointipolitiikka on satunnainen varastopaikan valinta. Luokkapohjainen varastointipolitiikkaa vaatii enemmän tilaa kuin satunnainen, mutta vähemmän kuin dedikoitu. Varastointipolitiikan vaikutusta kulkumatkaan niin automaatti- kuin manuaalivarastoissa on tutkittu useissa tutkimuksissa. Satunnaiseen varastopaikan valintaan verrattuna CBS-politiikka lyhentää kulkumatkaa, mutta se on yleensä pidempi kuin jos käytettäisiin dedikoituja varastopaikkoja. [7]

4.2 Keruustrategia

Kuten kappaleessa 2.3 on mainittu, keräily on kustannuksiltaan kallein toiminto varastossa. Se on myös monimutkaisin toiminto varastossa, sillä keräysprosessin ohjaamiseen liittyy useita päätöksiä, kuten kuvassa 4.2 on esitetty. Järjestelmäominaisuudet ovat jo varaston suunnitteluvaiheessa päätettäviä tekijöitä, joiden pohjalta itse operatiiviset ohjauspolitiikat päätetään. Seuraavissa osioissa on kuvattu tarkemmin, mitä päätöksiä keräysprosessin ohjaukseen liittyy. [6, 7]



Kuva 4.2: Keruuprosessiin vaikuttavat tekijät (Mukaillen [7])

4.2.1 Niputus

Yksinkertaisin tapa tehdä keräilyä on kerätä yksi tilaus kerrallaan. Tällöin kerääjä kerää yhden tilauksen sisältämät nimikkeet yhdellä keruumatkalla. Keräilyä voidaan kuitenkin tehostaa keräämällä useampi tilaus samalla kertaa, mikä lyhentää keräilymatkaa. Tätä useamman tilauksen yhdistämistä yhdellä keruumatkalla kerättäväksi kutsutaan tilausten niputukseksi ja syntynyttä tilausjoukkoa nipuksi. Niputusta käytettäessä syntyy tarve lajitella tilaukset pakkaamista varten. Lajittelu voidaan tehdä jo kerätessä, esimerkiksi keräämällä jokainen tilaus erilliseen yksikköön, tai vasta keräilyn jälkeen. [5]

Niputusta tehdessä pyritään minimoimaan tilausten keräämiseen tarvittava työ, joka yleensä tarkoittaa keruumatkan minimoimista. Rajoitteina toimivat esimerkiksi nipun paino, tilavuus, tilausten määrä nipussa, nipun arvioitu keruu-aika tai jokin näiden yhdistelmä. Julkaistut tutkimukset niputuksesta ovat pääosin keskittyneet kahden eri heuristiikan tutkimiseen. Ne molemmat hyödyntävät läheisyys-metriikkaa, joka kertoo kuinka lähellä kaksi tilausta on keruumatkalla. Tunnettuja läheisyys-metriikoita on useita. [5]

Monimutkaiseksi niputuksen tekee se, että tilauksilla on yleensä tietty määräaika, johon mennessä se pitää olla kerättynä. Määräaika määräytyy varastolta lähtevän kuljetuksen mukaan, johon tilaus tulee ehtiä. Myös tämä seikka tulee ottaa huomioon niputusta suunniteltaessa.

4.2.2 Reititys

Reitityksellä tarkoitetaan järjestyksen ja reitin valintaa kerättäessä tietyistä varastopaikoista. Ongelma on tarkasteltavalle varastolle ominainen kauppamatkustajan ongelma. Suorakulmion muotoiselle varastolle, jossa ei ole keskikäytävää, voidaan ratkaista optimaalinen reitti dynaamisen ohjelmoinnin avulla lineaarisessa ajassa. Myös varastoja, joilla on keskikäytävä, on tarkasteltu kirjallisuudessa ja niillekin voidaan ratkaista optimaalinen reitti. [7]

Optimaalinen reitti aiheuttaa kuitenkin ongelmia käytännössä. Reitti voi olla hyvin epälooginen kerääjälle, minkä johdosta he poikkeavat reitiltä. Optimaalinen reitti riippuu jättöpaikan sijainnista sekä varaston geometriasta, minkä vuoksi optimaalista reittiä ei voida laskea kaikissa varastoissa. Optimaalisen reitin laskeminen on raskasta, mikä voi aiheuttaa hitautta varastonohjausjärjestelmän toiminnassa. Optimaalinen reititys ei myöskään huomioi käytävien ruuhkautumista. Näistä syistä johtuen käytännössä reititys tehdään jollakin seuraavista heuristiikoista. [7] Kirjallisuudessa on myös todettu, että heuristiset menetelmät tuottavat lähes optimaalisen reitin, joka on ei ole käyttäjälle hämmentävä. [12]

Yksinkertainen reititysheuristiikka on S-reititys. Heuristiikan muodostama reitti on lähellä S-kirjainta, mistä heuristiikka on saanut nimensäkin. Käytävä, jolta täytyy kerätä jotain, kuljetaan kokonaan toiseen päähän. Käytävillä, joilta ei tarvitse kerätä mitään, ei mennä. [7]

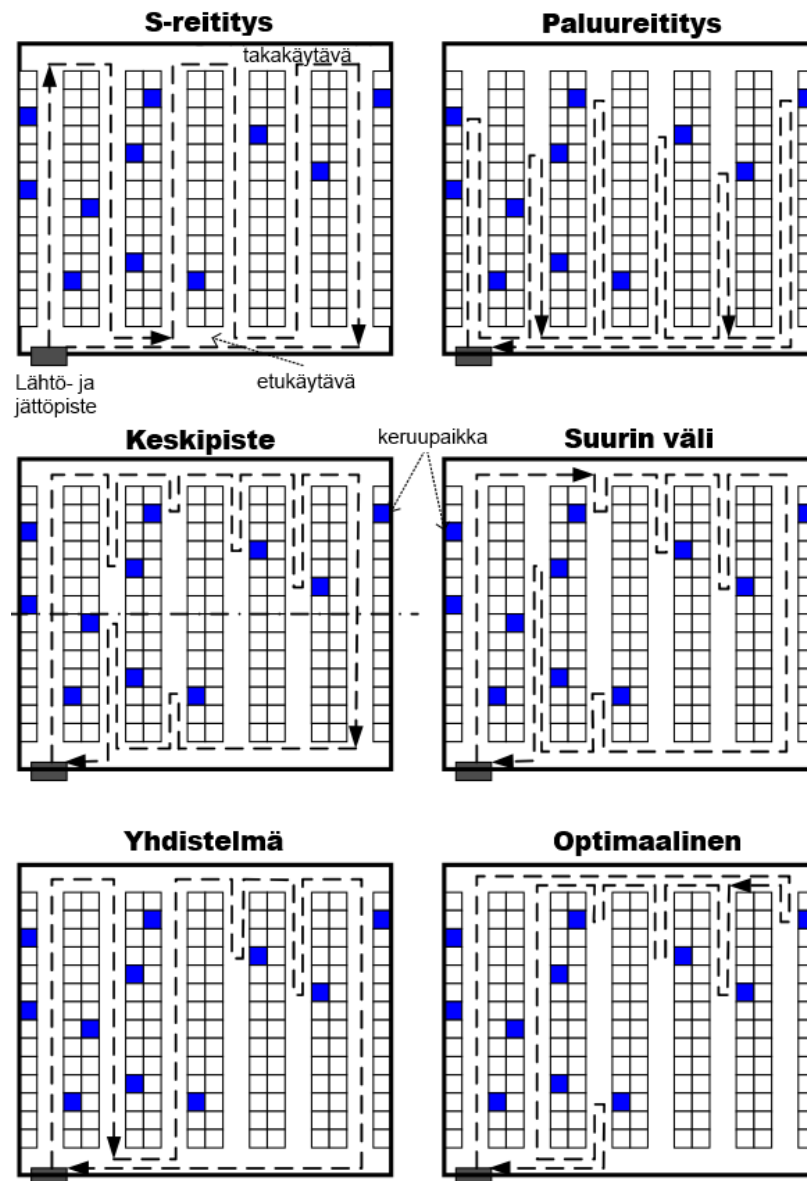
Paluuheuristiikassa kerääjä menee ja lähtee käytävältä aina samasta päästä. Myös tässä heuristiikassa kerääjä käy vain käytävillä, joilla on kerättävää. [7]

Keskipisteheuristiikka jakaa varaston kahteen osaan, joissa kummassakin käytetään paluuheuristiikkaa. Kerääjä kulkee toiseen osaan joko ensimmäisessä tai viimeisessä käytävältä. Kuvassa 4.3 on annettu esimerkkireitti, jonka keskipisteheuristiikka antaisi kuvan mukaisilla keruupaikoilla. [7]

Suurin väli -heuristiikka on samantyyppinen kuin keskipisteheuristiikka. Sen sijaan, että kerääjä ei menisi keskipistettä pidemmälle, kerääjä menee käytävällä suurimmalle välille asti. Suurin väli voi olla joko kahden keruupaikan välillä, jolloin kerääjä tekee paluureitin kummassakin päässä käytävää, tai keruupaikan ja etu-/takakäytävän välillä, jolloin kerääjä tekee paluureitin vain toisessa päässä käytävää. Suurin väli -heuristiikan antama esimerkkireitti on esitetty kuvassa 4.3. [7]

Yhdistelmäheuristiikka pyrkii yhdistämään S-reitityksen ja paluuheuristiikan parhaat puolet. Heuristiikka päättää jokaisen käytävän kohdalla, jolla on keruupaikkoja,

kuljetaanko käytävä kokonaan vai palataanko takaisin lähtöpäähän. Valinta tehdään hyödyntämällä dynaamista ohjelmointia. Esimerkki heuristiikan antamasta reitistä on esitetty kuvassa 4.3. [7]



Kuva 4.3: Havainnollistus eri reititysheuristiikoista (Mukaillen [7])

4.2.3 Keruualueet

Varasto voidaan jakaa useampaan keruualueeseen, joilla on omat kerääjänsä. Aluejako voidaan tehdä joko progressiivisena tai synkronoituna. Progressiivisessa aluejaossa erä (joka voi olla esimerkiksi yksi tilaus) etenee alueelta alueelle. Erän keräämistä tietyllä alueella ei aloiteta, ennen kuin se on kokonaisuudessaan kerätty aiemmalta alueelta. Synkronoidussa aluejaossa kerääjät taas toimivat rinnakkain ja keräävät samaa

erää yhtä aikaa. Seuraavan erän keräämistä ei aloiteta, ennen kuin aiempi erä on kerätty kaikilta alueilta. Tämä estää erien sekoittumisen. [7] Synkronoitua aluejakoa käytettäessä tulee ottaa huomioon myös tilausten jakaantuminen eri alueille.

Keruualueen jakamisen useampaan alueeseen suurin hyöty on se, että se vähentää kuljettua matkaa sekä käytävien ruuhkautumista. [7] Useamman eri keruualueen käyttäminen vaatii kuitenkin yhdistelyprosessin, mikä lisää työmäärää ja monimutkaistaa kokonaisprosessia. Yhdistely tarvitaan, jotta saman tilauksen eri alueilta tulevat materiaalivirrat saadaan yhdistettyä samaan asiakkaalle lähtevään yksikköön. Mikäli yhdistelyä ei tehdä, lähtevät eri alueilta tulevat tuotteet asiakkaalle eri yksiköissä, mikä lisää rahtikustannuksia. Yleensä yhdistely tehdään pakkauksen yhteydessä.

5. VARASTOPROSESSIEN MALLINNUS JA SIMULOINTI

Mallintaminen on tapa ratkaista reaalimaailman ongelmia. Sitä hyödynnetään etenkin, kun kokeet tosielämässä ovat kalliita tai mahdottomia. Mallintamisessa reaalimaailman ongelma tai järjestelmä abstrahoidaan malliksi, minkä jälkeen mallia voi analysoida ja optimoida. Lopuksi malli siirretään takaisin reaalimaailmaan. [13]

Mallit voidaan jakaa analyyttisiin malleihin ja simulaatiomalleihin. Analyyttinen malli on kuvaus syötteistä tuloksiin. Analyyttisiä malleja kutsutaan myös staattisiksi malleiksi. Simulointimalli, eli dynaaminen malli, on joukko sääntöjä, joka kuvaavat miten mallinnettava järjestelmä käyttäytyy ajan kanssa. Simuloinnissa järjestelmän tila muuttuu ajan kanssa joko diskreetisti tai jatkuvasti. [13]

Mallinnus tehdään aina jollakin abstraktiotasolla. Abstraktiotaso kuvaa, kuinka yksityiskohtaisesti järjestelmää tarkastellaan. Se voi vaihdella yksittäisten fyysisten objektien mallintamisesta aina kokonaisen väestön mallintamiseen. Tarkasteltavasta järjestelmästä riippuu, millä abstraktiotasolla mallinnus kannattaa tehdä. Esimerkiksi mallinnettaessa säätöjärjestelmää tehdään järjestelmän tarkastelu yksittäisten objektien tasolla, kun taas mallinnettaessa esimerkiksi liikennettä, tarkastellaan liikennevirtoja, eikä yksittäisiä ajoneuvoja. [13]

Kolme yleisintä mallinnukseen käytettyä simulointimenetelmää ovat differentiaaliyhtälömalli (engl. System Dynamics, SD), tapahtumapohjainen malli (engl. Discrete Events, DE) ja agenttipohjainen malli (engl. Agent Based Modeling, ABM). Eri menetelmät soveltuvat erilailla eri abstraktiotasoille. Differentiaaliyhtälömalleissa käytetään aggregaattiarvoja, minkä vuoksi se soveltuu hyvin korkean abstraktiotason mallintamiseen. Vastaavasti tapahtumapohjaisessa -menetelmässä tarkastellaan yksittäisiä olioita ja resursseja, minkä vuoksi se soveltuu alemman abstraktiotason mallintamiseen. [13]

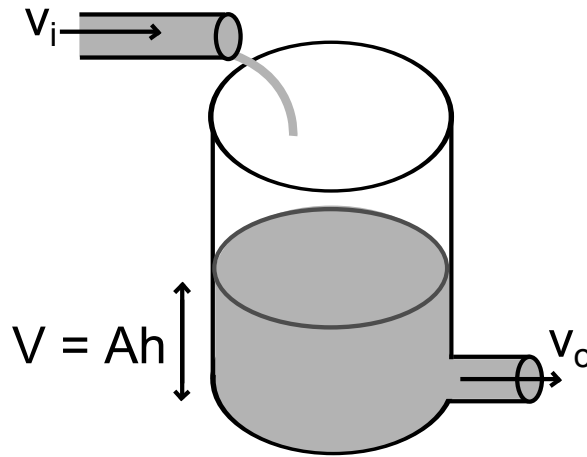
Uusin menetelmistä on agenttipohjainen mallinnus, jossa tarkasteltava järjestelmä mallinnetaan itsenäisiä, hajautettuna agentteina. Agenttipohjainen simulointi on hyvin skaalautuvaa, minkä vuoksi sitä käytetään kaikilla abstraktiotasoilla. [13]

5.1 Differentiaaliyhtälömalli

Differentiaaliyhtälömalleja käytetään monimutkaisten järjestelmien käyttäytymisen mallintamiseen. Menetelmässä järjestelmän osia tarkastellaan korkealla abstraktiota-

solla aggregaattiarvoina, eikä yksittäisinä tapahtumina tai yksilöinä. Tarkasteltavana oleva järjestelmä mallinnetaan aggregaattiarvoja kuvaavina säiliöinä ja niiden välisinä virtoina. Säiliö voi olla esimerkiksi varaston saldo ja virrat varastoon saapuvia ja lähteviä tavaravirtoja. Matemaattisesti differentiaaliyhtälömalli kuvaa järjestelmän nimensä mukaisesti joukkona differentiaaliyhtälöitä. [13]

Yksinkertainen esimerkki differentiaaliyhtälömallista on nestesäiliön pinnankorkeuden mallinnus, jota on havainnoillistettu kuvassa 5.1. Jos säiliöön virtaa nestettä vakionopeudella v_i ja nestettä ulos nopeudella v_o , on nesteen määrän muutos \dot{V} säiliössä näiden erotus $\dot{V} = v_i - v_o$. Ulosvirtausnopeus v_o ei kuitenkaan ole vakio, vaan riippuu pinnankorkeudesta. Bernoullin yhtälön mukaan nesteen ulosvirtausnopeus v_o voidaan kirjoittaa $v_o = \sqrt{2gh}$, missä g on gravitaatiovakio ja h pinnankorkeus. Jos säiliön oletetaan lieriön muotoinen, voidaan nesteen määrän muutos säiliössä kirjoittaa $\dot{V} = A\dot{h}$, missä A on säiliön pohjan pinta-ala ja \dot{h} pinnankorkeuden muutosnopeus. Tällöin nesteen pinnankorkeuden malli on $A\dot{h} = v_i - \sqrt{2gh}$. [14]



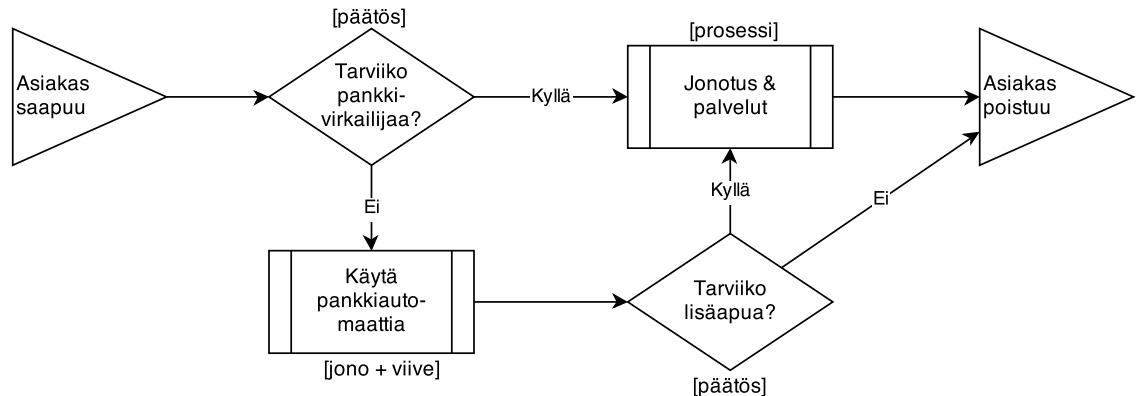
Kuva 5.1: Nestesäiliön pinnankorkeuden mallinnus

5.2 Tapahtumapohjainen malli

Tapahtumapohjainen mallinnus on menetelmä, jossa mallinnus tapahtuu sarjana tapahtumia, jotka tapahtuvat tietyllä ajanhetkellä ja siirtävät järjestelmän tilasta toiseen. Järjestelmä kuvataan matalalla abstraktiotasolla itsenäisinä olioina, resursseina ja lohkokaavioina. Oliot voivat olla esimerkiksi ihmisiä, tuotteita tai viestejä. Lohkokaaviot kuvaavat miten oliot liikkuvat järjestelmässä ja käyttävät resursseja. Olio voi olla esimerkiksi lavallinen tuotteita, joka liikkuu varastoprosessin läpi vastaanotosta varastoon ja sieltä keräyksen kautta lähettämöön. [13]

Esimerkkinä tapahtumapohjaisesta mallista voidaan mainita pankin jonotuksen mallintaminen. Tämän esimerkkimallin lohkokaavio on esitetty kuvassa 5.2. Saapuvat asiakkaat ovat olioita, jotka asioivat joko pankkiautomaatilla, pankkivirkailijoiden

luona tai molemmilla. Asiointi pankkiautomaatilla on mallinnettu jonotuksena automaatille sekä itse asiointista, joka on mallinnettu viiveenä. Virkailijat ovat mallin resursseja, joita on rajallinen määrä. Asiointi virkailijoiden luona koostuu jonotuksesta sekä erilaisista asiointiprosesseista. [13]



Kuva 5.2: Pankkiasioinnin tapahtumapohjainenmalli (mukaillen [13])

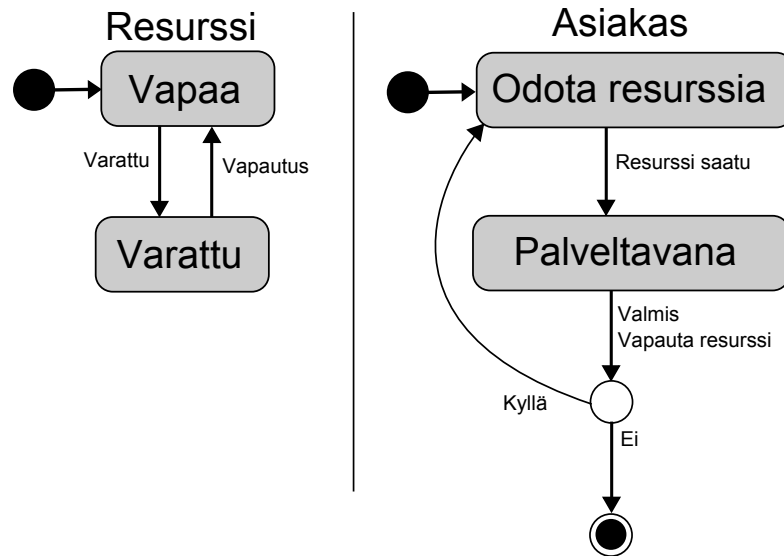
5.3 Agenttipohjainen mallinnus

Agenttipohjaisessa mallinnuksessa järjestelmä mallinnetaan joukkona itsenäisinä, päätöksiä tekevinä olioina, joita kutsutaan agenteiksi. Jokaisella agentilla on joukko sääntöjä, joiden pohjalta se saadaan käyttäytymään tietyllä, halutulla tavalla. Itse simulointi tapahtuu toistamalla agenttien välistä kanssakäymisiä. Jo hyvin yksinkertaisella mallilla, jossa järjestelmä on kuvattu joukkona agentteja ja niiden välille määritettyjä käyttäytymissääntöjä, päästään hyvin monimutkaisiin käyttäytymismalleihin, joita ei voida kuvata perinteisillä menetelmillä [15]

Bonabeau [15] esittää kolme etua, jotka agenttipohjaisella mallintamisella on muihin menetelmiin nähden. Ensimmäinen näistä on se, että ABM:n avulla on mahdollista havaita ja tunnistaa ilmiöt, jotka ovat seurausta yksilöiden välisestä käyttäytymisestä (engl. emergent phenomena). Nämä ilmiöt ovat usein intuition vastaisia, minkä vuoksi niiden ennustaminen on hankalaa. Koska koko ABM:n idea on mallintaa järjestelmä sen yksittäisinä osina, soveltuu se luontevasti tällaisten ilmiöiden havaitsemiseen.

Toinen etu, mikä ABM-menetelmällä on, on sen luontainen tapa kuvata mallinnettava järjestelmä. Yksittäisille osille annettavat säännöt tekevät mallista helpommin ymmärrettävän ja intuitiivisemman, kuin jos se kuvattaisiin yhtälöillä. Sääntöjen luomiseen voidaan hyödyntää olemassa olevia tutkimuksia ja dataa, jolloin agentit saadaan käyttäytymään hyvinkin todenmukaisesti. [15]

Yksinkertaisena esimerkkinä agenttipohjaisesta mallintamisesta on esitetty jonotusprosessin mallinnus, jonka tilakaavio on esitetty kuvassa 5.3. Agentteina mallissa ovat resurssit (esimerkiksi pankkivirkailijat) sekä asiakkaat. Resurssit palvelevat asiakkaita. Asiakkaat odottavat palveltavaksi pääsemistä. Palvelun jälkeen asiakas joko poistuu tai siirtyy jonottamaan uutta resurssia.



Kuva 5.3: Resurssien jonotuksen mallinnus agenttimenetelmällä (mukaillen [13])

5.4 Varastoprosessien simulointi

Yleisin käyttökohde varastoprosessien simuloinnille on ohjausstrategioiden testaus ja vertailu. Ohjaus formuloidaan useimmiten joko heuristisesti, jolloin se perustuu joukkoon sääntöjä, tai analyyttisesti, joilloin se perustuu tietyillä oletuksilla saatuun analyyttiseen kaavaan. Heuristinen ohjaus voi esimerkiksi antaa jokaiselle nimikkeelle täsmälleen saman verran varastopaikkoja. Analyyttinen ohjaus taas voi esimerkiksi olla kaava, jolla pystytään laskemaan nimikkeille varastopaikkojen määrät, kun kysynnästä oletetaan jotain.

Moni kirjallisuudessa toteutettu erilaisten varastoprosessien ohjausmenetelmien vertailu perustuu differentiaaliyhtälömalliin, mutta myös tapahtumapohjaisiin ja agenttipohjaisiin malleihin perustuvia simulointeja on toteutettu. Differentiaaliyhtälömallilla oletetaan yleensä parametreilta tietyt arvot, jotka voivat olla joko vakioita, tai ne voidaan saada jakaumasta. Tämän jälkeen voidaan laskea jokin aggregaattiarvo, kuten esimerkiksi kokonaistyömäärä, ja verrata sitä muiden ohjausmenetelmien tuloksiin.

Simuloitaessa varastoprosesseja tapahtumapohjaisilla malleilla tarkastellaan yleensä tarkemmin yksittäisiä olioita, kuten tuotteita, sekä resursseja, kuten työntekijöitä

tai trukkeja. Resursseilta oletetaan yleensä jotain, kuten esimerkiksi kävely- ja tuotteiden käsittelynopeus. Syötteinä toimivat esimerkiksi asiakastilaukset. Agenttipohjaisissa varastoprosessisimulaatioissa jokainen olio mallinnetaan agenttina, joka toimii itsenäisesti, mutta vuorovaikuttaa muiden agenttien kanssa. Esimerkiksi keräilyä tekevä työntekijä voi olla oma agentti, samoin kuin asiakas, joka lähettää tilauksia. Tällöin myös keräilyn ohjaus voi olla oma agentti, joka luo tiettyjen sääntöjen perusteella tilausten pohjalta keräilytyöt.

Varastoprosessit ovat luonteeltaan hyvin operatiivisia, joissa käsitellään esimerkiksi resursseja, määriä ja ajoitusta. Esimerkkinä voi mainita keräilyprosessin, jossa oikea määrä oikeita tuotteita pitää saada oikeaan aikaan kerättyä varastosta, jotta se ehtii lähtevään kuljetukseen. Varastoissa käsitellään fyysisiä tuotteita, joiden koko ja sijainti ovat tiedossa. Tuotteiden käsittely tapahtuu resurssien toimeista, joita ovat joko automaatio tai ihmistyöntekijät. Voidaankin siis sanoa, että varastoprosessit ovat matalalla abstraktiotasolla. Kuten kappaleen alussa on kuvattu, soveltuvat tapahtumapohjaiset ja agenttipohjaiset mallit parhaiten tällaisten matalan abstraktiotason ongelmien mallintamiseen. Seuraavassa on esitetty muutamia esimerkkejä siitä, miten varastoprosessien simulointi on toteutettu kirjallisuudessa.

Gagliardi et al. toteuttivat diskreetteihin tapahtumiin perustuvan simulaattorin mm. perunalastuja valmistavan tehtaan lopputuotevaraston toiminnan analysointiin. Simulaattorin tavoitteena oli tarjota työkalu varastointi- ja keruustrategioiden vertailuun. Työssä simuloitiin keräilyä lavojen läpivirtaushyllyistä koostuvalla aktiivialueella, josta keruu tehdään vieressä olevalle kuljettimelle. Aktiivialueella käytettiin dedikoituja varastopaikkoja. Simulaation parametreina olivat aktiivialueen varastopaikkajako, kerääjän kävely- ja tuotteiden käsittelyaika sekä täytön suoritus aika. Syötteenä simulaatiolle olivat asiakkaiden tilaukset, jossa hyödynnettiin logistiikka-keskuksen historiadataa. Simulaatiossa vertailtiin neljää heuristista ohjaussääntöä jakaa aktiivialueen varastopaikat tuotteille. [16]

Triebig et al. toteuttivat SeSAM-simulointityökalulla agenttipohjaisen simulaation korkeavaraston toiminnasta. Muun muassa kuljettimet, lukijat ja itse hyllystöhissit mallinnettiin itsenäisinä agentteina. Simulaattoria hyödynnettiin korkeavaraston ohjauksen testauksessa, kun varsinainen korkeavarasto oli vielä rakenteilla. [17]

Ito et al. rakensivat Javalla agenttipohjaisen mallin varastosta osana toimitusketjua. Malli koostui kolmesta osajärjestelmästä. Kommunikaatiojärjestelmä vastaa kommunikoinnista toimittajien ja asiakkaiden kanssa välittämällä tilauksia ja materiaaleja. Se koostuu toimittaja- ja asiakasagentista. Saldon hallinnoinnista ja ohjauksesta vastaava osajärjestelmä varaa asiakkaalle lähetettävän saldon ja luo tilaukset tavaran toimittajille. Se koostuu tilaus-, tuote-, saldo- ja toimittajatilausagenteista. Materiaalien käsittelystä vastaava osajärjestelmä siirtää tuotteet varastoon ja varastosta vihivaunuja (engl. AGV, automated guided vehicle) ohjaamalla. [18]

Macro et al. kehittivät diskreetteihin tapahtumiin perustuvan simulaatiomallin ProModel-ohjelmistolla varaston kapasiteetin ja täyttöasteen vertailuun ja analysointiin erilaisilla hyllykonfiguraatioilla. Malli ottaa syötteenään asiakastilaukset, varastosaldon ja tuotantotilaukset. Se koostuu olioista, sijainneista ja resursseista. Oliot ohjaavat toimintoja järjestelmässä ja niitä ovat esimerkiksi tuotepakkaus, lava ja asiakastilaus. Toiminnot tapahtuvat sijainneissa ja niihin tarvitaan tietyt resurssit. Sijainteja ovat mm. tuotantosolu ja lavavarasto hyllyineen. Resursseja taas ovat kerääjät, trukit ja trukkikukset. Mallin logiikka kuvattiin lohkoakaavioilla. Mallia hyödynnettiin kahden tuotantolaitoksen lopputuotevaraston tulevaisuuden varastotilan riittävyyden analysoimisessa. Syötteenä käytettiin historiatiedoista saatuja tilauksia. [19]

6. AKTIIVI- JA RESERVIALUE

Varastoinnin tavoitteena on tehokas tilankäyttö. Keräilyn tavoitteena taas on tilausten osoittamien nimikkeiden nopea noutaminen varastosta. Nämä tavoitteet ovat ristiriidassa keskenään, sillä nimikkeiden varastointi tiiviisti tekee niistä vaikeammin kerättäviä. Tämän ristiriidan ratkaisemiseksi moni varasto on jaettu aktiivi- ja reservialueeseen. Aktiivialue on joukko varastopaikkoja, joista tuotteiden kerääminen onnistuu helposti ja nopeasti, kuten esimerkiksi pienellä ja tiiviillä alueella sijaitsevat läpivirtaushyllyt. Aktiivipaikkojen vähäinen määrä vähentää kulkumatkaa keräiltäessä ja tehostaa täten keräilyä. Reservialueen reservipaikat ovat vaikeammin käsiteltävissä olevia varastopaikkoja, jotka ovat tilankäytöltään tiiviitä, kuten esimerkiksi korkeat kuormalavahyllyt. Reservipaikoille varastoidaan suurin osa tuotteista, josta niitä siirretään tarvittaessa aktiivipaikoille. [20]

Aktiivialueen luonteen vuoksi niihin mahtuu rajallinen määrä tuotteita. Kun tietty nimike loppuu aktiivialueelta, täytyy sitä tuoda lisää reservialueelta. Tätä toimenpidettä kutsutaan täytöksi. Täyttö itsessään on turha toimenpide, sillä siinä joudutaan käsittelemään jo kerran varastoituja nimikkeitä. Täyttöjä kuitenkin tarvitaan, jotta aktiivialuetta pystytään käyttämään. Mitä enemmän aktiivipaikkoja on, sitä vähemmän tarvitaan täyttöjä aktiivialueen ylläpitämiseksi, mutta sitä hitaampaa keräily aktiivialueelta on. Aktiivipaikkojen ja täyttöjen määrällä onkin käänteinen riippuvuus. [20, 21]

Aktiivialuetta käsittelevät tutkimukset voidaan jakaa karkeasti kolmeen ryhmään: aktiivialueen suunnitteluun keskittyneisiin, aktiivialueen käyttöön keskittyneisiin sekä täyttöihin keskittyneisiin tutkimuksiin. Aktiivialueen suunnitteluun keskittyneissä tutkimuksissa tarkastellaan miten aktiivi- ja reservialueet kannattaa mitoittaa asetettujen rajoitteiden puitteissa. Aktiivialueen käyttöön keskittyneiden tutkimusten keskeinen tutkimusongelma on valita mitä nimikkeitä aktiivialueelle varastoidaan ja miten paljon, jotta tehty työmäärä minimoituu. Täyttöihin keskittyneissä tutkimuksissa pyritään priorisoimaan täyttöjärjestystä tai minimoimaan täyttöjen määrää. Seuraavissa kappaleissa on tarkasteltu aktiivialueen käyttöön ja täyttöihin keskittyntä kirjallisuutta. Aktiivialueen mitoittamiseen keskittyneet tutkimukset on rajattu tarkastelun ulkopuolelle, sillä mitoitus on enemmänkin varaston suunnitteluvaiheessa tehtävä päätös, johon ei varaston ohjausta suunniteltaessa enää voi vaikuttaa.

6.1 Aktiivialueiden luokittelu

Varastoissa käsitellään erikokoisia tuotteita. Suurimpia nimikkeitä mahtuu vain yksi per lava, jolloin ne sekä varastoidaan että myös kerätään kokonaisina lavoina. [4] Suurilla nimikkeillä ei yleensä tarvita erillistä aktiivialuetta, koska ne käsitellään aina kokonaisena lavana, jota siirretään trukilla. Lavan siirtäminen korkealta hyllyltä ei eroa merkittävästi alemmalta hyllyltä siirtämisestä.

Tuotepakkauksella tarkoitetaan tässä yhteydessä suorakulmaista pahvilaatikkoa, joka painaa kahdesta kahteenkymmeneen kiloon, ja jota yksi henkilö pystyy käsittelemään ilman apuvälineitä. Pakkaus voi tässä yhteydessä olla yksittäinen tuote tai useamman tuotteen sisältävä tukkupakkaus. Ne varastoidaan yleensä lavoilla, mutta keräys sen sijaan tehdään yksittäisinä pakkauksina. Yksittäisten pakkauksen kerääminen on huomattavasti helpompaa lattialla olevalta lavalta kuin korkealla hyllyssä olevalta lavalta. Tämän takia pakkauksille varasto jaetaan aktiivi- ja reservialueeseen, missä aktiivialueelta kerääminen on huomattavasti helpompaa kuin reservialueelta. Koska pakkaukset varastoidaan lavoilla, myös täyttö tapahtuu aina kokonaisen lavan siirtämisellä, eli yleensä trukkiäsiirtona. Tällaista aktiivialuetta kutsutaan tässä työssä lavapohjaiseksi aktiivialueeksi. [4]

Keskipokoisia tuotteita pienemmät tuotteet ovat aina yksittäisiä tuotteita. Niiden käsittely on työläintä ja ei yleensä onnistu hyvin automaatiolta. Yhä suurempi osa logistiikkakeskusten tavarankäsittelystä on juuri pienempiä tuotteita, koska asiakkaat tilaavat pienempiä määriä ja useammin. Pienet tuotteet kerätään yleensä pahvi- tai muovilaatikoista, jotka on varastoitu suoraan hyllylle. Hylly on yleensä joko pientavarahylly tai läpivirtaushylly. Myös pienille tuotteille varasto jaetaan yleensä aktiivi- ja reservialueeseen. Pienillä tuotteilla täyttöä ei voida tehdä puhtaana lavasiirtona. Tällaista aktiivialuetta kutsutaan tässä työssä hyllypohjaiseksi aktiivialueeksi. [4]

6.1.1 Hyllypohjaisen aktiivialueen ohjaus

Ensimmäisen matemaattisen kuvauksen aktiivi-reservipaikkaongelmasta antoivat Hackman ja Rosenblatt. [22] He hyödynsivät nestemallia, joka olettaa nimikkeiden olevan jaettavissa mielivaltaisiin tilavuuksiin, joka käytännössä vaatii, että tarkasteltavien nimikkeiden ovat fyysiseltä kooltaan pieniä. Lisäksi heidän mallinsa olettaa, että nimikkeen täytön suorittamiseen kuluva aika on riippumaton täytettävästä määrästä. He kuvasivat aktiivipaikoille sijoitettavien nimikkeiden ja niiden määrän valinnan matemaattisena optimointiongelmana seuraavien muuttujin.

s	=	säästö, kun keruu tehdään aktiivista reservin sijaan
c_r	=	täyttöön kuluva aika
p_i	=	nimikkeen i keruutöiden määrä tarkasteluvälillä
f_i	=	nimikkeen i kysyntä tilavuutena
v_i	=	nimikkeelle i varattava tilavuus aktiivista
$f_i(v_i)$	=	nimikkeen i säästö, kun sitä varastoivaan tilavuus v_i
N	=	nimikkeiden määrä
V	=	aktiivialueen kokonaistilavuus

Hyödyntämällä nesteen virtausmallia voidaan täyttöjen määrä tarkasteluaanjaksosona arvioida. Jos nimikettä i virtaa varaston läpi f_i kuutiometriä ja sitä varastoidaan aktiivialueella v_i kuutiometrin verran, tarvitaan tarkasteluaanjaksosona suunnilleen f_i/v_i täyttöä. [22] Tällöin, kun $a_i = sp_i$ ja $b_i = c_r f_i$, nimikkeen i säästö on

$$f(v_i) = \begin{cases} sp_i - \frac{c_r f_i}{v_i} = a_i - \frac{b_i}{v_i} & \text{jos } v_i > 0 \\ 0 & \text{jos } v_i = 0 \end{cases} \quad (6.1)$$

Funktiosta voidaan ratkaista pienin järkevä määrä, jossa nimikettä varastoidaan aktiivialueelle. Mikäli nimikettä i varastoitaisiin alle määrän $\frac{c_r f_i}{sp_i}$ olisi nimikkeen tarjoama säästö negatiivinen. [22]

Ratkaisu aktiivipaikoille sijoitettavista nimikkeistä saadaan ratkaisemalla ensin osaongelma ”kuinka paljon valittuja nimikkeitä kannattaa varastoida aktiivialueella”. Tämä voidaan esittää seuraavana optimointiongelmana, kun S on joukko aktiiviin varastoitavia nimikkeitä. [22]

$$\begin{aligned} & \arg \max_{v_i} \sum_{i \in S} f(v_i) \\ \text{s.e.} \quad & \sum_{i \in S} v_i \leq V \\ & v_i > 0, \quad i \in S \end{aligned} \quad (6.2)$$

Nyt tavoitefunktio $f(v_i)$ on jatkuva, aidosti kasvava ja derivoitava koko määrittelyalueellaan, joten ratkaisussa voidaan hyödyntää Karush-Kuhn-Tucker optimaalisuusehtoja. Hackman ja Rosenblatt johtivat näitä hyödyntämällä seuraavan lausekkeen, jolla nimikkeille S voidaan laskea optimaalinen määrä v_i^* . [22]

$$v_i^* = \left(\frac{\sqrt{b_i}}{\sum_{j \in S} \sqrt{b_j}} \right) = \left(\frac{\sqrt{f_i}}{\sum_{j \in S} \sqrt{f_j}} \right) \quad (6.3)$$

Jotta saavutettava säästö maksimoituu, täytyy aktiivialueelle varastoitavat nimikkeet varastoida lausekkeen 6.3 antamissa määrissä. Tätä hyödyntämällä voidaan aktiivialueelle varastoitavien nimikkeiden valinta muodostaa nyt seuraavaksi opti-

mointiongelmaksiksi, missä $x_i \in \{0, 1\}$ on päätösmuuttuja, joka kuvaa varastoidaanko nimikettä i aktiiviin. [22]

$$\begin{aligned} \arg \max_{x_i} \quad & \sum_{i=1}^N f(v_i^*) x_i \\ \text{s.e.} \quad & \sum_{i=1}^N x_i v_i^* \leq V \\ & x_i \in \{0, 1\} \end{aligned} \tag{6.4}$$

Yhtälöistä havaitaan, että ongelma on tyypiltään reppuongelma (engl. knapsack problem). Reppuongelmassa tuotteita, joilla jokaisella on tietty paino ja arvo, pakataa reppuun. Tuotteet pyritään valitsemaan siten, että kokonaisarvo maksimoituu ilman, että painolle asetettua rajaa ei ylitetä. Reppuongelma on tunnetusti NP-vaikea, eli sitä ei pysty ratkaisemaan polynomiajassa. Reppuongelman ratkaisemiseksi on kuitenkin olemassa hyvin tunnettu ahne heuristiikka, joka antaa lähellä optimia olevan ratkaisu. Ahne heuristiikka valitsee aina kulloinkin parhaan paikallisen vaihtoehdon ja pyrkii täten optimitulokseen. Tuotteet järjestetään yksikköarvon mukaiseen järjestykseen ja ratkaisujoukkoon valitaan järjestyksessä jokainen tuote, joka vielä mahtuu. Yleinen ahne algoritmi reppuongelman ratkaisemiseksi on esitetty algoritmossa 1. [23]

Algoritmi 1: Ahne heuristiikka reppuongelman ratkaisemiseksi

Syötteen: Tuotteiden painot w_i

Tuotteiden arvot p_i

Repun koko V

Tulokset: Valitut tuotteet S

```

1 Järjestä tuotteet yksikköarvon  $p_i/w_i$  mukaan laskevaan järjestykseen
2  $S := \emptyset$ 
3 for  $i = 1, 2, \dots, N$  do
4   if  $V > w_i$  then
5      $V := V - w_i$ 
6      $S := S \cup \{i\}$ 
7   end
8 end
```

Ongelma 6.4 pystytään ratkaisemaan ahneella heuristiikalla. Nimikkeet järjestetään järjestykseen niiden tarjoaman hyödyn, eli yksikköarvon, mukaan ja aktiivialue täytetään tässä järjestyksessä. Tämän ongelman tapauksessa nimikkeen yksikköarvo on [22]

$$\frac{f(v_i^*)}{v_i^*} = \frac{a_i}{\sqrt{b_i}} \sum_{j \in S} \sqrt{b_j} - \left(\sum_{j \in S} \sqrt{b_j} \right)^2. \tag{6.5}$$

Lausekkeesta 6.5 havaitaan, että nimikkeiden välinen järjestys on riippumaton aktiiviin sijoitettavasta nimikejoukosta S . Tällöin yksikköarvo yksinkertaistuu muotoon

$$y_i = \frac{a_i}{\sqrt{b_i}} = \frac{p_i}{\sqrt{f_i}}. \quad (6.6)$$

Hackmanin ja Rosenblattin esittämä ahne heuristinen algoritmi on esitetty algoritmissa 2. [22]

Algoritmi 2: Nestemallin heuristinen algoritmi, joka valitsee aktiivialueelle varastoitavat nimikkeet siten, että kokonaistyömäärä (keräykset + täytöt) minimoituu.

Syötteet : Keruutöiden määrät nimikkeittäin p_i
 Nimikkeiden menekki tilavuutena f_i
 Säästö keskimäärin kun kerätään aktiivista eikä reservistä s
 Täyttöön keskimäärin kuluva aika c_r

Tulokset : Aktiiviin varastoitavat nimikejoukko S_k
 Nimikkeille varattu tilavuus aktiivista v_i^*

```

1 for  $i = 1, 2, \dots, N$  do
2   | Laske nimikkeelle  $i$  yksikköarvo  $y_i$  kaavalla 6.6
3 end
4 Järjestä nimikkeet laskevaan järjestykseen yksikköarvon  $y_i$  mukaan
5 // Käy nimikkeet läpi yksikköarvojen  $y_i$  mukaisessa järjestyksessä:
6  $S_0 := \emptyset$ 
7 for  $k = 1, 2, \dots, N$  do
8   |  $S_k := S_{k-1} \cup \{k\}$ 
9   | // Laske nimikejoukolle  $S_k$  kokonaissäästö  $a_k$  ja sen nimikkeille määrät  $v_{ki}$ 
10  |  $a_k := 0$ 
11  | foreach  $i \in S_k$  do
12    | Laske nimikkeelle  $i$  varastoitava määrä  $v_{ki}$  lausekkeella 6.3
13    | Laske nimikkeen  $i$  tuoma säästö  $f(v_{ki})$  lausekkeella 6.1
14    |  $a_k := a_k + f(v_{ki})$ 
15  | end
16 end
17 Valitse aktiiviin varastoitavaksi nimikejoukko  $S_k$ , jonka tarjoama säästö  $a_k$  on
   | suurin
18 foreach  $i \in S_k$  do
19   | Laske nimikkeelle  $i$  varastoitava määrä  $v_i^*$  lausekkeella 6.3
20 end

```

Bartholdi ja Hackman vertailivat Hackmanin ja Rosenblattin optimaalista aktiivipaikoille nimikkeiden määrän valintaa kahteen yleisesti teollisuudessa käytössä olevaan aktiivipaikkojen varastointistrategiaan: saman verran tilaa jokaiselle nimikkeelle ja jokaista nimikkeitä sen verran, että sitä riittää aktiivissa yhtä pitkäksi aikaa ennen loppumista. [24] He näyttivät, että optimaalisella sijoituksella täytöt jakau-

tuvat tasaisesti koko aktiivialueelle. He osoittivat, että mitä enemmän nimikkeiden menekit eroavat toisistaan, sitä suurempi hyöty saadaan täyttämällä aktiivipaikat optimaalisesta. He suorittivat myös case-tutkimuksen lääketukkurin jakelukeskuksessa, jossa aktiivialue koostui 325 kartongeille tarkoitetuista läpivirtaushyllystä. He ennustivat, että optimaalisella aktiivipaikkajaolla täyttöjen määrä vähenisi 40 prosenttia, tai vastaavasti 20 prosenttia, mikäli aktiivialueesta käytettäisiin vain 285 hyllyä.

Gu ym. osoittivat, että Hackmanin ja Rosenblattin esittämä algoritmi 2 ei anna kaikissa tapauksissa optimaalista ratkaisua. [25] He esittivät iteratiivisen branch-and-bound-algoritmin ongelman ratkaisemiseksi ja osoittivat, että algoritmi antaa optimiratkaisun kaikissa tapauksissa äärellisellä määrällä iteraatioita. He suorittivat laskennallisen vertailun heidän optimaalisen ratkaisun antavan algoritmin sekä Hackmanin ja Rosenblattin heuristiikan välillä. Heuristinen algoritmi saavutti optimiratkaisun suurimmassa osassa tapauksia ja niissä tapauksissa, joissa se ei sitä saavuttanut, oli ratkaisu kuitenkin erittäin lähellä optimiratkaisua.

Bloch ym. tarkastelivat aktiivi- ja reservipaikkaongelmaa yhdessä varastoinnin kanssa hyvin tiukoilla oletuksilla. [26] Heidän mallinsa minimioi täyttöihin ja keruuseen kuluneen kulkumatkan sijoittamalla jokaista nimikettä tasan yhteen paikkaan aktiivissa ja reservissä. He antoivat algoritmin mallin ratkaisemiseksi. Malli suoriutui kaikissa luoduissa testitapauksissa muita varastointipolitiikkoja paremmin, mutta oletusten vuoksi malli soveltuu kuitenkin vain hyvin tarkasti rajattuihin tapauksiin: täyttö tapahtuu kerääjän toimesta heti kun tuote loppuu aktiivista, varastorakenne on tietynlainen, keräily tehdään käyttämällä S-reititystä ja keruutöiden määrä tilausta kohden on vakio. Etenkin viimeinen oletus tekee mallista hyvin teoreettisen, sillä todellisuudessa tilausten keruumäärät vaihtelevat paljonkin.

Walter ym. huomauttivat kolme ongelmaa, jotka nestemallilla on [21]:

- Nimikkeet ovat reaali maailmassa diskreetin kokoisia, jolloin tilan mielivaltainen jakaminen on vain approksimaatio todellisuudesta.
- Usein nimikkeiden määrää aktiivipaikoilla ei voida kasvattaa vain yhden tuotteen verran, vaan useampi tuote kerralla. Esimerkiksi läpivirtaushyllyllä yksi kaista on usein varattu yhdelle nimikkeelle, jolloin nimikkeen määrää aktiivipaikoilla kannattaa kasvattaa kaistan eikä yhden nimikkeen verran.
- Joitakin nimikkeitä, kuten esimerkiksi ämpäreitä, voidaan säilöä sisäkkäin, jolloin kaksi nimikettä vie todellisuudessa vähemmän tilaa kuin niiden tilavuuksien summa.

Walter ym. käsittelevät aktiivi-reservipaikkaongelmaa jatkuvan mallin sijaan diskreetillä mallilla, joka huomioi nimikkeiden fyysiset dimensiot. [21] He tutkivat kolmea

ongelmaa: (1) kuinka paljon valittuja nimikkeitä kannattaa sijoittaa aktiivipaikoille, (2) mitä nimikkeitä kannattaa sijoittaa aktiiviin ja kuinka paljon sekä (3) minkä kokoinen aktiivialue kannattaa varata valituille nimikkeille. Jokaista ongelmaa tarkasteltiin kahdella parametrijoukolla: yhdellä, jossa nimikettä voidaan varastoida minkä tahansa kokoisissa hyllyissä, sekä toisella, jossa jokaisella nimikkeellä on tasan yksi hyllykoko, jossa sitä varastoidaan.

Walter ym. osoittavat, että ongelma (1) on vaihtelevan kokoisilla hyllyillä NP-kova, mutta vakiokokoisilla ekvivalentti diskreetin resursointiongelman kanssa, joka pystytään ratkaisemaan polynomiajassa. [21] He myös osoittavat, että ongelman (1) jatkuva relaksaatio on yleistys Hackmanin ja Rosenblattin nestemallista ja esittävät neljä korjausheuristiikkaa, joilla nestemallin antama ratkaisu saadaan muutettua diskreetiksi. He esittävät laskennallisen vertailun korjausheuristiikkojen antamista tuloksista diskreetin mallin optimaaliseen ratkaisuun, joka osoittaa, että oikeilla korjausheuristiikoilla päästään riittävän lähelle optimiratkaisua.

Walter ym. osoittavat ongelman (2) olevan sekä vaihtelevan kokoisilla että vakiokokoisilla hyllyillä NP-kova. [21] Vakiokokoisien hyllyjen tapauksessa ongelma voidaan kuitenkin ratkaista ratkaisemalla ongelma (1) kaikille nimikevalinnoille, joka kuitenkin johtaa eksponentiaaliseen suoritusaikaan. Heidän pienellä aineistoilla suorittama laskennallinen vertailu osoittaa nestemallin ja parhaan korjausheuristiikan yhdistelmän pääsevän riittävän lähelle diskreettiä optimiratkaisua.

Bartholdi ja Hackman laajentavat Hackmanin ja Rosenblattin mallia. [4] He näyttivät, miten mallia voidaan käyttää tuoteryhmien kanssa, eli kun tietyt nimikkeet pitää varastoida yhdessä. Hackmanin ja Rosenblattin malli olettaa, että täyttö suoritetaan kun nimike loppuu aktiivipaikoilta. Bartholdi ja Hackman laajensivat mallin toimimaan myös tapauksissa, joissa täyttö suoritetaan kun nimikkeen määrä aktiivipaikoilla tippuu alle tietyn, määrätyn rajan. He määrittelevät lausekkeen, jolla voidaan päätellä kannattaako nimike sijoittaa kokonaan aktiivi- tai reservipaikoille. He laajensivat mallia myös huomioimaan, mikäli nimikettä ei vielä ole aktiivipaikoilla tai se pitää siirtää kokonaan pois. Lisäksi he esittävät heuristiikan, millä voidaan valita nimikkeet ja niiden määrät, mikäli aktiivialueita on useampi ja kaikilla on oma keruusäästö.

6.1.2 Lavapohjaisen aktiivialueen ohjaus

Lavapohjaiselle aktiivialueelle on tyypillistä, että varastointi ja täytöt tapahtuvat kokonaisina lavoina, mutta keräily yksittäisinä pakkauksina. Bartholdi ja Hackman tarkastelevat miten pakkausten keruualue kannattaa järjestää. [4] He esittävät teoreeman siitä, miten paljon nimikettä kannattaa aktiivialueelle varastoida.

Teoreema 6.1.1 *Nimikettä kannattaa varastoida aktiivialueelle joko ei yhtään, pienin*

järkevä määrä tai kaikki.

Pienin järkevä määrä on yksi lava, ellei jostain erityisestä syystä, esimerkki käytävien ruuhkautumisen vähentämiseksi, nimikettä haluta varastoida useampaa lavaa. Tämä johtuu siitä, että pienimmän järkevän määrän jälkeen nimikemäärän kasvattaminen aktiivipaikoilla ei lisää keruutöiden määrää eikä vähennä täyttöjä, paitsi jos nimikkeen koko saldo sijoitetaan aktiivipaikoille. [4]

Olkoon nimikkeelle i p_i keruutehtävien määrä, joissa kerättävä määrä on alle lavamäärän, d_i lavojen kokonaismäärä, joita tyhjenee kyseisillä keruilla ja D_i kuinka monta lavaa kokolavakeruutehtävät siirtävät. Olkoon l_i haluttu paikkojen minimimäärä ja u_i maksimimäärä aktiivialueella. Olkoon keskimääräisesti yhden aktiivikeruutehtävän suorittaminen s sekuntia nopeampaa kuin yhden reservikeruutehtävän ja täyttöön keskimääräisesti kuluva aika c_r sekuntia. Tällöin kokonaissäästö, kun nimikkeelle i annetaan x aktiivipaikkaa, on [4]

$$\begin{cases} 0 & \text{jos } x = 0 \\ sp_i - c_r d_i & \text{jos } l_i < x < u_i \\ s(p_i + D_i) & \text{jos } x = u_i \end{cases} \quad (6.7)$$

Lausekkeesta (6.7) voidaan havaita, että tiettyjä nimikkeitä kannattaa varastoida aktiiviin vain täysi määrä u_i , tai ei ollenkaan. Muussa tapauksessa kokonaissäästö on negatiivinen. Näin käy, kun keskimääräinen kerättävä määrä d_i/p_i on suurempi kuin suhde s/c_r . [4]

Nimikkeiden valinnasta lavapohjaiselle aktiivialueelle voidaan muodostaa optimointiongelma. Muuttujalla $x_i \in \{0, 1\}$ kuvataan, varastoidaanko nimikettä i aktiivialueelle ja V on aktiivipaikkojen kokonaislukumäärä. Tavoitefunktio on kokonaistyömäärä (keruut + täytöt), joka pyritään minimoimaan. Kun c_1 on keskimääräinen keruukustannus aktiivista ja c_2 reservistä, voidaan ongelma kuvata seuraavasti. [4]

$$\begin{aligned} \arg \min_{x_i} \quad & \sum_{i=1}^N \left[(c_1 p_i + c_r d_i) x_i + c_2 p_i (1 - x_i) \right] \\ \text{s.e.} \quad & \sum_{i=1}^N l_i x_i \leq V \\ & x_i \in \{0, 1\} \end{aligned} \quad (6.8)$$

Toteamalla $s = c_2 - c_1$ ja kääntämällä ongelma maksimointiongelmaksi voidaan se saattaa muotoon

$$\begin{aligned}
& \arg \max_{x_i} \sum_{i=1}^N (sp_i + c_r d_i) x_i \\
& \text{s.e.} \quad \sum_{i=1}^N l_i x_i \leq V \\
& \quad \quad \quad x_i \in \{0, 1\}
\end{aligned} \tag{6.9}$$

Tästä yhtälömuodosta havaitaan, että ongelma eräs tapaus reppuongelmasta. Kuten aiemmin on todettu, reppuongelma on tunnetusti NP-vaikea, eli sitä ei pysty ratkaisemaan polynomiajassa. Kyseiselle ongelmalle saadaan kuitenkin riittävän lähellä optimia oleva ratkaisu käyttämällä ahnetta menetelmää, joka on esitetty algoritmissa 1. Nimikkeet järjestetään järjestykseen niiden yksikköarvon mukaan ja aktiivialue täytetään tässä järjestyksessä. Tämän ongelman tapauksessa nimikkeen yksikköarvo on työvoimasäästö per aktiivipaikka (engl. labor-savings per forward location), jonka Bartholdi ja Hackman nimesivät työtehokkuudeksi (engl. labor efficiency): [4]

$$\frac{(sp_i - c_r d_i)}{l_i} \tag{6.10}$$

Nimikkeen suuri menekki ei vielä takaa sitä, että sen työtehokkuus on suuri. Esimerkiksi jos nimikkeen tilattu määrä per tilaus on keskimäärin puoli lavallista, olisi nimikkeellä keskimäärin 2 keruuta aktiivista, minkä jälkeen sitä pitäisi jo täyttää. Jos keruu aktiivista säästää keskimäärin minuutin ja täyttö vie kolme minuuttia, olisi kokonaissäästö negatiivinen. Tällaisten korkean menekin ja voluumin nimikkeiden kohdalla tulee harkita, pitäisikö ne sijoittaa kokonaisuudessaan aktiiviin tai massapaikoille.

Nimikkeelle i voidaan harkita sen varastoimista kokonaisuudessaan aktiiviin, mikäli sen lähiajan saldon yläraja on tiedossa. Tällöin nimike kerätään kokonaisuudessaan aktiivista ja sen kokonaissäästö on

$$s(p_i + D_i). \tag{6.11}$$

Ja säästö per varastopaikka on tällöin

$$s(p_i + D_i)/u_i. \tag{6.12}$$

Bartholdi ja Hackman esittävät teoreeman, jonka mukaan nimike kannattaa varastoida kokonaisuudessaan aktiiviin, mikäli sen tarjoama säästö per varastopaikka on suurempi kuin nimikkeen työtehokkuus, joka on esitetty lausekkeessa 6.10. [4]

Hyödyntämällä lausekkeitä 6.10 ja 6.12 voidaan ongelma mallintaa optimointiongelmana, jolla saadaan ratkaistua mitä nimikkeitä kannattaa varastoida aktiivialueella

ja miten paljon. Muuttujalla $z_j \in \{0, 1\}$ kuvataan varastoidaanko nimikkeen kaikki u_j yksikköä kokonaisuudessaan aktiiviin, jolloin saavutetaan säästö $s(p_i + D_i)$. Nämä ovat nimikkeitä, joita ei kannata varastoida aktiivialueelle vähempää kuin kaikki. Muuttujalla $x_i \in \{0, 1\}$ taas kuvataan varastoidaanko nimikettä i sen minimimäärä l_i aktiiviin, jolloin saavutetaan säästö $(sp_i - c_r d_i)$. Vastaavasti muuttujalla y_i kuvataan varastoidaanko nimikkeen i loput $u_i - l_i$ yksikköä aktiiviin, jolloin saavutettava säästö on $sD_i + c_r d_i$. Jäljelle jäävät $u_i - l_i$ yksikköä voidaan kuitenkin varastoida vasta, kun minimimäärä l_i on varastoitu, mikä varmistetaan rajoitteella $y_i \leq x_i$. Ongelma voidaan nyt formalisoida optimointiongelmaksiksi. [4]

$$\begin{aligned}
 & \arg \max_{x_i, y_i, z_j} \left[\sum_i \left[(sp_i - c_r d_i)x_i + (sD_i + c_r d_i)y_i \right] + \sum_j s(p_j + D_j)z_j \right] \\
 & \text{s.e.} \quad \sum_i (l_i x_i + (u_i - l_i)y_i) + \sum_j u_j z_j \leq V \\
 & \quad \quad \quad x_i, y_i, z_j \in \{0, 1\} \\
 & \quad \quad \quad y_i \leq x_i
 \end{aligned} \tag{6.13}$$

Ongelma on muuten identtinen reppuongelman kanssa, mutta siinä on ylimääräinen rajoite $y_i \leq x_i$. Bartholdi ja Hackman [4] osoittavat, että ahneen heuristiikan antama ratkaisu ongelmaan 6.13 on käypä, lähellä optimia ja toteuttaa ylimääräisen rajoitteen $y_i \leq x_i$.

Bartholdi ja Hackman esittävät ahneen heuristiikan lähellä optimia olevan ratkaisun löytämiseksi. Kyseinen heuristiikka on kuvattu algoritmossa 3. Heuristiikka voidaan kuvitella olevan eräänlainen huutokauppa, jossa jokaisella nimikkeellä on arvo, millä se kilpailee aktiivin varastopaikoista. Nimikkeet jaetaan niihin, jotka kannattaa varastoida kokonaisuudessaan aktiiviin (eli ne, joiden tarjoama säästö per varastopaikka 6.12 on suurempi kuin työtehokkuus 6.10) sekä muihin nimikkeisiin. Kokonaan aktiiviin varastoitavat nimikkeet osallistuvat u_i varastopaikan voittamiseksi arvolla $s(p_i + D_i)/u_i$. Muut nimikkeet osallistuvat ensin l_i varastopaikan voittamiseksi arvolla $(sp_i - c_r d_i)/l_i$. Jos nimike voittaa l_i paikkaa aktiivista, osallistuu se tämän jälkeen huutokauppaan $u_i - l_i$ paikan voittamiseksi arvolla $(sD_i + c_r d_i)/(u_i - l_i)$. Tätä huutokauppaa jatkamalla jaetaan kaikki aktiivialueen varastopaikat nimikkeille.

6.2 Täytöt

Van den Berg ym. argumentoivat, että koska useissa varastoissa täytöt tapahtuvat yksikkö kerrallaan (esimerkiksi lava kerrallaan), ei yhtä tuotetta kannata varastoida aktiivipaikoilla yhtä yksikköä enempää. [27] He tarkastelevat varastoprosessia, jossa keräily tehdään erillisenä keruuajana ja ennen tätä on aikaa täyttää aktiivipaikat

Algoritmi 3: Lavapohjaisen mallin ahne heuristinen algoritmi, joka valitsee varastoitavat nimikkeet siten, että kokonaistyömäärä (keräykset + täytöt) minimoituu.

Syötteet : Alle lavamäärän keruutehtävien määrä nimikkeittäin p_i
 Alle lavamäärän keruulla tyhjenevien lavojen määrä nimikkeittäin d_i
 Täyden lavamäärän keruutehtävien määrä nimikkeittäin D_i
 Minimimäärä lavapaikkoja nimikkeittäin l_i
 Maksimimäärä lavapaikkoja nimikkeittäin u_i
 Säästö keskimäärin kun kerätään aktiivista eikä reservistä s
 Täyttöön keskimäärin kuluva aika c_r
 Aktiivipaikkojen määrä V

Tulokset : Lavapaikkojen määrä aktiivialueelta nimikkeittäin x_i

```

1 // Lasketaan nimikkeille arvo  $b_i$ , jolla ne osallistuvat huutokauppaan
2 // ja varastopaikkojen määrä  $v_i$ , josta ne kilpailevat
3 for  $i = 1, 2, \dots, N$  do
4   Laske nimikkeelle  $i$  säästö per varastopaikka  $spv_i$  kaavalla 6.12
5   Laske nimikkeelle  $i$  työtehokkuus  $t_i$  kaavalla 6.10
6   if  $spv_i \leq t_i$  then
7      $b_i := (sp_i - c_r d_i) / l_i$ 
8      $v_i := l_i$ 
9   else
10     $b_i := s(p_i + D_i) / u_i$ 
11     $v_i := u_i$ 
12  end
13 end
14 while Tilaa aktiivialueella do
15   Valitse nimike  $i$ , jonka arvo  $b_i$  on suurin
16   // Anna nimikkeelle  $i$  tilaa  $v_i$  varastopaikan verran
17    $x_i := v_i$ 
18   // Laske nimikkeelle uusi arvo, jolla se osallistuu huutokauppaan
19   if  $v_i = u_i$  then
20     // Nimikkeelle  $i$  on varattu maksimimäärä paikkoja
21     Poista nimike  $i$  huutokaupasta
22   else
23     // Nimikkeelle  $i$  on varattu  $l_i$  paikkaa,
24     // nimike kilpailee vielä  $u_i - l_i$  lisäpaikasta
25      $b_i := (sD_i + c_r d_i) / (u_i - l_i)$ 
26      $v_i := u_i$ 
27   end
28 end
```

erillisenä täyttöaikana. Tavoitteena on valita täytöt, jotka minimoivat työmäärän keruuajana. He muotoilevat ongelman binäärioptimointiongelmana ja esittävät ahneen heuristiikan sen ratkaisemiseksi. He osoittavat, että heuristiikan antama

ratkaisu on käypä ja tietyn arvon sisällä optimista. He vertailivat heuristiikkaa pelkkään reservivarastointiin sekä kahteen yleisesti teollisuudessa käytössä olevaan aktiivipaikkojen varastointistrategiaan: saman verran tilaa jokaiselle nimikkeelle ja jokaista nimikkeitä sen verran, että sitä riittää yhtä pitkäksi aikaa ennen loppumista. Heuristiikka pärjasi parhaiten kaikissa testitapauksissa.

Hollingsworth esitti diplomityössään mallin, jolla täyttöjen määrää voidaan vähentää ns. Dock-To-Forward-tekniikalla, eli varastoimalla nimikkeitä suoraan vastaanotosta aktiivipaikoille ilman että ne viedään ensin reserviin. [20] Hänen mallinsa olettaa, että tuotteilla ei ole dedikoituja paikkoja aktiivissa, täytöt ovat täysiä lavoja sekä tuotteiden saapuvat ja lähtevät toimitukset ovat tiedossa tarkasteluvälillä. Hän osoitti, että oletusten ollessa voimassa täyttöjen määrän minimoiminen on ekvivalentti sen kanssa, että minimoidaan aika, joka kestää kunnes aktiivipaikka vapautuu. Tämä aika on riippuvainen sekä nimikkeen kysynnästä että määrästä aktiivipaikoilla. Hän esitti ja vertaili neljää heuristiikkaa, jolla päätetään mitä nimikkeitä viedään aktiivipaikoille suoraan vastaanotosta sen jälkeen, kun tarkasteluvälin kysyntä on tyydytetty. Tulokset osoittivat, että Dock-To-Forward-tekniikan käyttö vähensi kokonaistäyttöjen määrää 18-24 prosenttia heuristiikasta riippuen.

De Vries ym. kehittivät mallin, jonka perusteella voidaan muodostaa prioriteettijärjestys tehtäville täytöille kun keräystä ja täyttöjä suoritetaan yhtä aikaa. [28] Malli olettaa, että tuotteiden lyhyen aikavälin kysyntä on tiedossa. He esittivät kolme täyttöpolitiikkaa, jotka pyrkivät minimoimaan nollakeruiden määrän (eli tuote päässyt loppumaan aktiivipaikoilta, kun sitä pitäisi kerätä). Tarkastelun kohteena olleessa varastossa keräys tehtiin aaltoina ja kaikki priorisointi tehtiin aina aallon aluksi. Ensimmäinen täyttöpolitiikka oli heuristinen ja siinä täyttöjen suoritusjärjestyksen määrää nimikkeen aktiivipaikoilla olevan saldon määrä suhteessa kysyntään. Toinen täyttöpolitiikka vaati lisäoletuksia: aallon pituus on tiedossa, täytön suoritusajankohdasta voidaan ennustaa, varastossa on saldoa riittävästi koko tilausten toimittamiseksi ja täytön jälkeen samaa tuotetta ei tarvitse enää täyttää saman aallon aikana. Näiden oletusten ollessa voimassa voidaan ongelma mallintaa binäärioptimointiongelmana ja sille saadaan nollakeruut minimoiva optimiratkaisu. Kolmas täyttöpolitiikka yksinkertaistaa toista olettamalla, että tuotteiden tilattu määrä jokaisella tilausrivillä on sama. Tämän oletuksen ollessa voimassa täyttöpolitiikka minimoi nollakeruiden määrän. De Vries ym. vertailivat esitettyjä täyttöpolitiikkoja toisiinsa ja satunnaisessa järjestyksessä tehtyihin täyttöihin. Kaikki täyttöpolitiikat suoriutuivat satunnaista täyttöjärjestystä paremmin.

Gagliardi ym. tarkastelivat prosessia, jossa keräys tehdään aktiivipaikoilta yksi tilaus kerrallaan suoraan kuljettimelle. [29] Heidän tutkimusongelmansa oli selvittää paljonko valittuja nimikkeitä kannattaa varastoidaan aktiivipaikoille ja millainen täyttöpolitiikka vähentää nollakeruiden määrää. He esittivät ja vertailivat kahta

heuristiikkaa nimikkeiden sijoittamiseksi aktiivipaikoille ja neljää täyttöjärjestyksen valinta -heuristiikkaa. He suorittivat simulaatiovertailun jakelukeskuksen kahdeksan viikon datalla. Vertailu osoitti, että heuristiikkoja hyödyntämällä nollakeruiden määrää pystyttiin selvästi vähentämään.

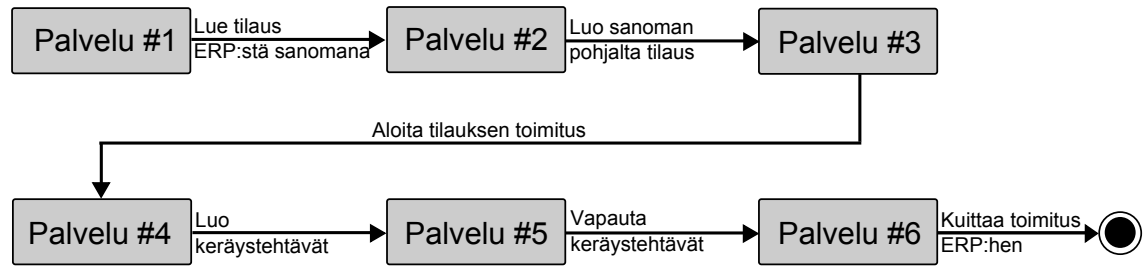
7. AKTIIVIALUEOHJAUS MOTUKSESSA

Varastoprosessien ohjaus Motuksessa tapahtuu usean eri palvelun toimesta. Palvelut ovat sovelluksia, jotka tunnistavat itsenäisesti milloin niiden täytyy suorittaa tietty toimenpide. Palveluiden toiminta perustuu pollaukseen, eli ne ajetaan tietyn ajan välein. Palvelu tarkistaa tietyn ajan välein, onko prosessi sellaisessa tilassa, että sen ohjaus on palvelun vastuulla. Mikäli prosessin ohjaus on siirtynyt palvelun vastuulle, päättää palvelu itsenäisesti mitä toimenpiteitä se tekee ja milloin. Toimenpiteet määräytyvät palveluun rakennetun logiikan ja erilaisten konfiguroitavien sääntöjen pohjalta. Palvelu suorittaa tarvittavat toimenpiteet, mikä siirtää prosessin seuraavaan tilaan, jolloin sen ohjaus siirtyy seuraavan palvelun vastuulle.

Palveluiden välillä ei ole suoraa kommunikaatiota, mikä takaa niiden riippumattomuuden toisista. Motuksen ohjaus muistuttaakin hyvin paljon agenttipohjaista mallia, missä palvelut ovat agenteja. Riippumattomien palveluiden käyttäminen tekee ohjauksesta hyvin vikasietoisen, sillä yhden palvelun pysähtyminen ei vielä pysäytä koko prosessia. Myös yhden palvelun korvaaminen toisella on helppoa, koska palveluiden välillä ei ole suoria riippuvuuksia. Haittapuolena on pollauksen käyttö, joka lisää viivettä prosessin toimintaan. Toiminnot eivät kuitenkaan ole kriittisiä ajan suhteen, jolloin pieni viive toiminnoissa ei haittaa kokonaisprosessin toimintaa. Viiveet eivät myöskään ole suuria, keskimäärin joitain kymmeniä sekunteja.

Esimerkkinä Motuksen suorittamasta ohjauksesta voi mainita tilauksen toimittamisen, joka on esitetty kuvassa 7.1. Palvelut suorittavat toimenpiteitä, jotka siirtävät ohjausvastuun seuraavalle palvelulle. Ensimmäinen palvelu kommunikoi toiminnohjausjärjestelmän kanssa siirtäen tilauksen sisältävän sanoman Motuksen puolelle. Seuraava palvelu jäsentää sanoman ja luo sen pohjalta tilauksen Motukseen. Seuraava palvelu päättää milloin tilauksen toimitus tulee aloittaa ja aloittaa toimituksen kun sen aika koittaa. Seuraava palvelu luo tilaukselle keräystehtävät. Seuraava palvelu vapauttaa keräystehtävät, eli päättää miltä alueelta tilauksen keräystehtävät suoritetaan ensin. Tämän jälkeen varastotyöntekijät voivat suorittaa tilauksen keräyksen ja pakkauksen. Kun tilaus on kerätty, pakattu ja luovutettu kuljetettavaksi, huolehtii viimeinen palvelu siitä, että kuittaus toimituksesta lähetetään toiminnohjausjärjestelmään.

Aktiivialueen ohjaus tapahtuu Motuksessa usean eri palvelun toimesta. Ohjaukseen pystytään vaikuttamaan nimikekohtaisten varastointi-, keräys- ja täyttösääntöjen



Kuva 7.1: Palveluiden ohjausvastuut tilauksen toimitusprosessissa

avulla, jotka on esitetty seuraavassa.

7.1 Varastointisäännöt

Motuksessa varastointia ohjataan varastointisääntöjen perusteella. Varastointisäännöt määrittävät minne nimikettä varastoidaan ja minkälaisina määrinä. Varastointikohteen voi määrittää joko yksittäisinä paikkoina, jolloin kyseessä on kiinteät varastopaikat (engl. dedicated storage), tai varastoalueena. Varastoalue on loogisesti ryhmitelty joukko varastopaikkoja, esimerkiksi aktiivipaikat. Varastoalueen sisällä millekkään nimikkeelle ei ole ennalta määrättyä varastopaikkaa, jolloin puhutaan dynaamisesta varastoinnista (engl. shared storage). Varsinaisen varastopaikan valintaan varastoalueen sisältä käytetään usein ABC-luokitusta. Varastointisäännöllä on määritetty maksimimäärä, joka kertoo paljonko nimikettä saa korkeintaan olla kohdepaikassa tai -alueella. Kohteessa oleva määrä plus varastoitava määrä ei siis saa ylittää maksimimäärää.

Varastointisääntöjä sovelletaan siinä vaiheessa, kun vastaanotetut nimikkeet halutaan varastoida. Ne käydään asetetussa järjestyksessä läpi ja ensimmäinen soveltuva sääntö määrää mihin varastoitavat nimikkeet siirretään.

Varastointisäännöt mahdollistavat erilaisten varastointiprosessien tukemisen. Niillä voidaan mallintaa esimerkiksi tilanne, jossa kaikki varastointi tehdään ensin reserviin, josta tuotteet siirretään aktiiviin kerättäväksi. Toisaalta ne mahdollistavat myös varastoinnin suoraan aktiivipaikoille. Se miten varastointisäännöt asetetaan riippuu pitkälti siitä, miten varastoprosesseja halutaan ohjata. Varastointisäännöt määritetään yleensä käsin tai kerralla kaikille nimikkeille.

7.2 Keräyssäännöt

Keräysprosessit määritetään Motukseen keräystapojen ja keräyssääntöjen avulla. Keräystapa kuvaa millä tavalla keräily tehdään. Esimerkiksi täyslavakeräily ja normaali, varastosta kohdeyksikköön tehtävä keräily ovat omia keräilytapoja.

Kun myyntitilausta lähdetään toimittamaan Motuksessa, varataan sille tilatun määrän verran saldoa varastosta. Nimikekohtaiset keräyssäännöt kuvaavat mistä eri

alueilta ja missä järjestyksessä saldoa pyritään varaamaan, sekä millä keräystavalla varatun saldon keräily tehdään. Näin voidaan mallintaa esimerkiksi tilanne, jossa ensisijaisesti kerätään kokonaisia lavoja reservialueelta, toissijaisesti kerätään tuotteita normaalilla tavalla aktiivialueelta ja lopuksi, mikäli kokonaisia lavoja ei reservistä löydy eikä aktiivissa ole vapaata saldoa, kerätään tuotteet reservistä.

Keräyssäännölle voidaan määrittää minimi- ja maksimimäärä, jolle keräyssääntöä sovelletaan. Esimerkiksi suuret määrät saattavat olla kätevämpiä kerätä reservialueelta aktiivialueen sijaan.

Keräyssäännöllä on myös ominaisuus, joka kuvaa saako se luoda täytön. Tämä tarkoittaa sitä, että jos keräyssäännön osoittamalta alueelta ei pystytä varaamaan riittävästi saldoa, voidaan sinne luoda täyttö, mikäli täyttösäännöt sen sallivat. Jos täyttö saatiin luotua, tehdään varaus tästä siirrettävästä saldoa. Keräyslupa näille saldovarauksille annetaan vasta, kun täyttö on suoritettu, mikä estää sen, että kerääjä joutuisi odottamaan täytön suorittamista. Tällä pystytään takaamaan se, että keräilyä ei edes tehdä reservialueelta, mikäli nimike on määritetty kerättäväksi aktiivialueelta.

7.3 Täyttösäännöt

Täyttöjä ohjataan Motuksessa täyttösääntöjen perusteella. Täyttösääntö kuvaa mistä (yleensä reservi) ja mihin (yleensä aktiivi) nimikettä siirretään. Täyttösääntö sisältää hälytysrajan sekä siirrettävän minimi- ja maksimimäärän. Hälytysraja on minimimäärä, joka tuotetta pyritään vähintään pitämään kohdealueella. Jos nimikkeen saldo kohdealueella tippuu alle hälytysrajan, luodaan nimikkeelle täyttö. Täyttösäännölle asetetaan myös keräystapa, jolla täyttö suoritetaan. Keräystapa voi olla esimerkiksi täyslavasiirto (esim. lavapohjaisella aktiivialueella) tai normaali keräily kohdelavalle, josta tuotteet hyllytetään aktiivialueelle (esim. hyllypohjainen aktiivialue).

Siirrettävä minimimäärä kuvaa, paljonko nimikettä pyritään vähintään siirtämään kerralla kohdealueelle. Minimimäärä on riittävä määrä, eli kun sen verran saldoa on varattu siirrettäväksi, ei enempää yritetä varata. Se ei kuitenkaan ole vaadittu määrä, eli jos siirrettäväksi varattu saldo jää alle minimimäärään, siirretään se mitä on pystytty varaamaan. Maksimimäärä taas kertoo, paljonko nimikettä saa korkeintaan siirtää kerralla.

Minimi- ja maksimimäärän lisäksi voidaan määrittää yksikkö-/varastopaikkakohtainen minimi- ja maksimiraja. Nämä kuvaavat paljonko yksikössä/varastopaikassa pitää minimissään ja saa korkeintaan olla, jotta sen saldoa harkitaan siirrettäväksi. Tällä voidaan taata se, että vähän tuotteita sisältäviä lavoja ei siirretä aktiivialueelle, mistä ne tyhjenisivät nopeasti.

7.4 Aktiivialueen mallintaminen

Vaikka Motuksessa ei itsessään ole käsitettä aktiivialue, pystytään se mallintamaan ja sitä ohjaamaan varastointi-, keräys- ja täyttösääntöjen avulla. Keräyssäännöillä ohjataan varataanko saldoa ensisijaisesti aktiivista vai reservistä. Täyttösäännöillä määritetään mille nimikkeille tehdään täyttöjä reservistä aktiiviin eli käytännössä mitkä nimikkeitä pidetään aktiivialueelle ja miten paljon. Varastointisäännöillä ohjataan vastaanotettuja nimikkeitä. Kaikki nimikkeet voidaan esimerkiksi ohjata suoraan reserviin, tai ohjata tietyt nimikkeet suoraan vastaanotosta aktiivialueelle.

Yksinkertaisimmillaan aktiivialueen ohjaamiseen riittää pelkkä keräys- ja täyttösääntöjen asettaminen. Keräyssäännöillä ohjataan saldon varaus ensisijaisesti aktiivialueelle ja täyttösäännöillä määritetään nimikkeet, joita aktiiviin halutaan varastoida. Jos varastointia tehdään pelkästään reserviin, eivät varastointisäännöt vaikuta aktiivialueen ohjaamiseen. Mutta kuten Hollingsworth työssään osoitti, voidaan suoralla varastoinnilla aktiivialueelle saavuttaa tehokkuusetuja. [20] Tällöin täytyy asettaa lisäksi myös varastointisäännöt aktiivialueelle varastoitaville nimikkeille.

Motus tukee kahdenlaista toimintapaa, mikäli saldoa aktiivialueelta ei enää pystytä varaamaan, mutta sitä on saatavilla reservistä. Varaus voidaan kohdistaa suoraan reserviin, tai aktiivikeräyssäännöille voidaan asettaa täyttöjen luonti päälle. Täytön luominen vähentää kokonaisuutena työmäärää, kuin jos keräily tehtäisiin reservistä, tai mikäli tilauksen muulle saldovaraukselle annettaisiin keräyslupa heti ja tilaus kerättäisiin aktiivialueelta kahtena tai useampana lenkinä. Se kuitenkin myös pidentää tilausten läpimenoaikaa.

7.4.1 Varastointi- ja täyttösääntöjen määrittäminen Motuksessa

Aiemmissa Motus-projekteissa aktiivialueen ohjaus on pohjautunut puhtaasti tuntuun ja kokemuseräiseen tietoon, eikä formaaleja menetelmiä ole käytetty. Ongelmaksi on noussut myös sääntöjen asettaminen. Jo pienissä varastoissa on usein useita satoja nimikkeitä, jolloin sääntöjen asettaminen kaikille nimikkeille käsin on työlästä. Tästä syystä johtuen tietyillä asiakkailla täyttösääntöjä ei ole luotu ollenkaan eikä aktiivialueen käyttöä ole oikeastaan ohjattu Motuksen toimesta ollenkaan. Toisilla asiakkailla täyttösäännöt on määritetty käsin työnjohdon toimesta yritys ja erehdysmenetelmällä, mikä on ollut työlästä ja aikaa vievää.

Tätä varten Motuksessa on olemassa eräs täyttösääntöjen automaattinen laskentamenetelmä, joka on kuvattu algoritmissa 4. Laskenta pohjautuu yksinkertaiseen heuristiikkaan, jossa tarkastellaan nimikkeen ottokertoja. Ottokerta tarkoittaa yhtä keräystehtävää, jossa nimikettä on kerätty yksiköstä tai varastopaikasta. Algoritmi toimii seuraavalla tavalla. Jokaiselle nimikkeelle määritetään historiatiedoista paljonko nimikettä on yleensä vastaanotettu per yksikkö. Tämä kertoo paljonko on täysi

yksiköllinen nimikettä. Jokaiselle nimikkeelle määritetään ABC-luokka sen perusteella, montako erillistä keruutapahtumaa, eli ottokertaa, kyseisellä nimikkeellä on valitulla tarkasteluvälillä ollut. Tarkasteluväli voidaan määrittää asiakaskohtaisesti. Jokaiselle luokalle A, B ja C on määritetty hälytysraja sekä minimi täyttömäärä täyden yksikön prosenttiosuutena. Nimikkeille luodaan täyttösäännöt sen perusteella mihin luokkaan se kuuluu ja montako tuotetta täysi yksiköllinen nimikettä on. [30]

Algoritmi 4: Motuksessa nykyisin käytössä oleva menetelmä täyttösääntöjen automaattiseksi määrittämiseksi.

Syötteet : Nimikkeiden vastaanotetut määrät v_i

Nimikkeiden ottokerrat tarkasteluvälillä p_i

Nimikkeiden prosentuaalinen osuus A, B ja C-luokassa q_a, q_b, q_c

ABC-luokakohtaiset hälytysrajat ja minimitäyttömäärät

Tulokset : Täyttösäännöt nimikkeittäin

- 1 Määritä nimikkeille vastaanotetuista määristä, paljonko nimikettä on yleensä per lava
 - 2 Määritä nimikkeille ABC-luokka ottokertojen perusteella. Luokkaan A q_a prosenttia nimikkeistä, luokkaan B q_b ja luokkaan C q_c
 - 3 Luo jokaisen luokan A, B ja C nimikkeille täyttösäännöt annettujen hälytysraja- ja minimitäyttömäärien suhteen.
-

Yleensä täytöt tehdään kokonaisina lavoina, jolloin täyttö pystytään tekemään trukki siirtona ilman nimikkeiden siirtämistä lavalta toiselle. Tällöin lavalla ei saa olla muita nimikkeitä. Tietyissä varastoissa on kuitenkin paljon sekalavoja, jotka sisältävät montaa eri nimikettä. Näissä varastoissa täyttö joudutaan tekemään keräilynä lavasiirron sijaan, eikä algoritmi 4 pystytä hyödyntämään.

Yleensä aktiivialueella halutaan pitää vain nopeimmin kiertäviä nimikkeitä, jolloin vain luokalle A asetetaan hälytysraja ja minimitäyttömäärä. Algoritmin ideana on, että menekiltään suurimpia nimikkeitä pidetään aktiivialueella. Intuitiivisesti tämä kuulostaa järkevältä, mutta kuten Bartholdi ja Hackman osoittivat, ei nimikkeen suuri menekki vielä takaa sitä, että nimikettä kannattaa varastoida aktiiviin vain yksi lava. [4] Tämä voi johtaa siihen, että aktiivialueen ylläpitämiseen tarvittavien täyttöihin kuluva työmäärä on suurempi kuin aktiivialueesta saatava hyöty.

Toinen heikkous, mikä tällä algoritmilla on, liittyy nimikkeen saldoihin. Ensinnäkään se ei ota huomioon aktiivialueen kokoa. Täyttösäännöt luodaan menekkiluokan mukaan, joka on ensisijaisesti tarkoitettu määräämään mihin varastopaikoille nimikkeet varastoalueen sisällä sijoitetaan. Täten esimerkiksi jos A-luokassa on nimikkeitä 800 kpl ja aktiivialueen koko on 2000 lavapaikkaa, jää 1200 paikkaa täyttämättä, jos jokaista nimikettä varastoidaan aktiivialueelle yksi lava. Toiseksi algoritmi ei huomioi onko nimikkeellä saldoa vai ei, jolloin täyttösääntöjä voidaan luoda nimikkeille, joita ei edes ole varastolla.

7.5 Varastointi- ja täyttösääntöjen rajoitteet

Motuksen täyttösäännöt on pyritty rakentamaan joustavasti, jotta ne tukisivat mahdollisimman erilaisia prosesseja. Täyttösääntöjen tarpeita voidaan tarkastella lavapohjaisen ja hyllypohjaisen aktiivialueen näkökulmasta. Lavapohjainen aktiivialue on esimerkiksi normaalin lavavaraston tietyt lattiapaikat sekä mahdollisesti heti niiden yläpuolella olevat kuormalavahyllyt. Hyllypohjainen aktiivialue koostuu hyllyistä, joihin nimikkeet varastoidaan joko suoraan tai todennäköisemmin pahvi- tai muovilaatikossa. Tällainen aktiivialue on esimerkiksi läpivirtaus- tai pientavarahyllyistä koostuva aktiivialue.

Lavapohjaisille aktiivialueille on ominaista, että täytöt tehdään kokonaisina lavoina. Jos varastossa on paljon useampaa eri nimikettä sisältäviä lavoja, on myös mahdollista että tuotteet kerätään kohdeyksikköön, joka lopulta varastoidaan aktiivialueelle. Lavapohjaisilla aktiivialueilla aktiivialueen koko määritetään varastopaikkojen määränä, eli käytännössä kuinka monta lavaa aktiivialueelle mahtuu. Tällöin luontevin tapa määritellä nimikkeelle kuinka paljon sitä pidetään aktiivialueella on lavojen määrä. Tässä tapauksessa ei myöskään tarvitse tietää, paljonko nimikettä lavalla on.

Hyllypohjaiselle aktiivialueelle on ominaista, että täytöt tehdään hyllytyksenä. Tämä tarkoittaa käytännössä sitä, että nimikkeet noudetaan reservistä esimerkiksi lavalla ja siirretään yksitellen aktiivialueelle hyllyyn. Aktiivialueen koko ei tällöin ole järkevä määrittää pelkästään hyllyjen määränä, koska riippuu puhtaasti nimikkeestä, paljonko sitä yhdelle hyllylle mahtuu. Luontevaa onkin määrittää aktiivialueen koko tilavuutena, joka saadaan esimerkiksi laskemalla yhteen kaikkien hyllyjen tilavuus. Nimikkeille taas luontevin tapa määrittää varastoitava määrä on kappaleina. Tämä voidaan laskea automaattisesti, kun nimikkeen mittadimensiot ovat tiedossa.

Käyttäjän kannalta olisi luontevinta määrittää nimikekohtaisesti paljonko nimikettä pidetään aktiivialueella ja milloin täyttö suoritetaan. Tämä toimintatapa, jossa nimikettä täytetään tiettyyn määrään asti, jos se tippuu tietyn rajan alle, tunnetaan ns. (s,S)-politiikkana. [28] Lavapohjaisella aktiivialueella määrä olisi luontevinta määrittää lavoina, kun taas hyllypohjaisella aktiivialueella määrä olisi luontevinta määrittää kappaleina nimikettä.

Motuksessa pystyy määrittämään hälytysrajan avulla minkä rajan jälkeen täyttö suoritetaan, mutta varsinaista maksimirajaa ei pysty määrittämään. Tämä johtuu siitä, että Motus ei huomioi hälytysrajaa tarkasteltaessa varattua saldoa, vaan laskee aktiivialueella jo olevan määrän aina vapaasta saldosta. Tällöin varsinaista maksimirajaa ei edes ole, jolloin nimikettä voi olla aktiivialueella paljonkin täyttösäännön siirretävää maksimimäärää enemmän. Syy, miksi Motus toimii näin, johtuu tilaus-ten saapumisesta. Tilauksille tehdään yksinkertaistettuna saldovaraus heti, kun se vastaanotetaan. Tällöin aikaisemmin saapunut tilaus saa varattua saldon aktiivis-

ta, vaikka sitä ei välttämättä kerätä ennen myöhemmin saapuneita tilauksia. Tällä toimintamallilla aktiivialueetta ei pysty kokonaisuudessaan jakamaan nimikkeille, koska näille ylimeneville määriille pitää jättää vapaata tilaa aktiivialueelta. Tällöin aktiivialueen tilankäyttö ei välttämättä ole kaikkein tehokkainta.

Jos nimikkeelle määritettäisiin oikea maksimiraja aktiivialueella, pitäisi saldovaraus pystyä tekemään tilauksille samassa järjestyksessä, kuin keräilykin tehdään. Tietyillä asetuksilla tämän pystyy jo nyt mahdollistamaan Motuksessa, mutta tähän liittyy ongelmia, jotka menevät tämän työn aiheen ulkopuolelle. Nimikkekohtainen aktiivialueen maksimiraja tehostaisi ainakin aktiivialueen tilankäyttöä, mutta saat-taisi nykyisellä toimintamallilla hidastaa keräilyä tietyissä tapauksissa. Toisaalta taas ongelma esiintyy todennäköisesti vain satunnaisesti tai hyvin nopeasti kiertävillä nimikkeillä, joille voi miettiä kannattaisiko niitä varastoida aktiivialueella enemmän.

Vaikka Motuksen täyttösäännöt eivät mahdollista lavapohjaisen aktiivialueen tapauksessa määrittää montako lavaa nimikettä pidetään aktiivialueella, pystytään niillä määrittämään, että aktiivialueelle siirretään maksimissaan vain yksi lava, kun saldo tippuu alle hälytysrajan. Tämä onnistuu asettamalla täyttösäännön keräilytavaksi täyslavasiirron ja minimirajan yhdeksi ja maksimirajan äärettömäksi. Vastaa-vasti myös nimikkeen kaikkien lavojen varastointi aktiivialueella onnistuu asettamalla sekä minimi- että maksimiraja äärettömäksi. Siirrettäväksi lavamäärän määrittäminen muuhun arvoon kuin yksi tai kaikki ei kuitenkaan onnistu. Tämä johtuu siitä, että yhden lavan sisältämien nimikkeiden määrä on harvemmin vakio. Suositeltavaa olisikin lisätä täyttösääntöihin nimikkeen minimi- ja maksimikappalemäärän rinnalle nimikkeen lavojen minimi- ja maksimimäärä. Tällöin pystyttäisiin määrittämään tarkemmin, montako lavallista nimikettä pidetään aktiivialueella.

Varastointisäännöillä määritetään maksimimäärä paljonko nimikettä saa olla kohdealueella. Varastointisäännöllä olisi hyvä olla myös maksimilavojen määrä, jolloin se soveltuisi selkeämmin myös lavapohjaisen aktiivialueen ohjaamiseen.

8. TÄYTTÖSÄÄNTÖJEN AUTOMAATTINEN MÄÄRITTÄMINEN

Kuten kappaleessa 6 on kuvattu, on aktiivialueen ohjaamiseen kehitetty muutamia formaaleja malleja. Kaksi lupaavinta mallia, joita voisi hyödyntää täyttösääntöjen automaattiseen määrittämiseen, ovat Hackmanin ja Rosenblattin 1990 ensimmäisenä esittämä nestemalli, sekä Bartholdin ja Hackmanin lavapohjaisen aktiivialueen ohjaamiseen esittämä lavapohjainen malli. Nestemalli on kuvattu kappaleessa 6.1.1 ja lavapohjainen malli on kuvattu kappaleessa 6.1.2.

Nstemalli olettaa tuotteiden olevan mielivaltaisiin osiin jaettavia, joten se soveltuu parhaiten pienikokoisia tuotteita sisältävälle aktiivialueelle, eli hyllypohjaiselle aktiivialueelle. Aktiivialue, jossa nestemallia voisi hyödyntää, voi esimerkiksi olla pientavara- tai läpivirtaushyllyistä koostuva aktiivialue. Lavapohjainen malli taas olettaa, että täyttö voidaan suorittaa lavasiirtona, jolloin se soveltuu luontevasti lavavaraston aktiivialueen ohjaamiseen. Tässä kappaleessa on pohdittu, miten nestemallia ja lavapohjaista mallia voisi hyödyntää täyttösääntöjen automaattisessa määrittämisessä Motukseen. Lopuksi esitetään menetelmät täyttösääntöjen määrittämiseksi sekä hylly- että lavapohjaisella aktiivialueella.

8.1 Mallien oletukset

Molemmissa malleissa on tiettyjä oletuksia ja on aina syytä miettiä tapauskohtaisesti, soveltuuko malli tarkasteltavaan prosessiin. Jos mallin oletukset eivät toteudu, eivät mallin antamat tulokset ole täysin optimaalisia. Toisaalta teoria on aina jonkinlainen yksinkertaistus reaalimaailmasta ja kaikki oletukset täyttyvät harvoin. Mallin antamat tulokset olla käyttökelpoisia tai vähintäänkin suuntaa antavia, kunhan tärkeimmät oletuksen täyttyvät.

Nstemallissa on käytetty seuraavia oletuksia.

- Täytön suorittamiseen kuluva aika on riippumaton täytettävästä määrästä. [22] Jos tämä ei toteudu, ei täyttöön kuluva aika c_r ole vakio. Käytännössä oletus tarkoittaa sitä, että täytettävän nimikkeen tuotteet on varastoitu lähikäin reserviin, jolloin sitä ei tarvitse hakea useasta eri varastopaikasta. Oletus toteutuu myös, jos täytettävät määrät ovat pieniä suhteessa määrään, jossa tuotetta varastoidaan reservissä, jolloin nimikkeitä ei tarvitse kerätä useammalta

varastopaikalta.

- Nimikettä aktiivialueelta yhdellä kertaa kerättävä määrä ei koskaan ylitä nimikkeen sille varattua tilaa. Lisäksi jos kerättävä määrä on suurempi kuin mitä nimikettä on aktiivissa, tehdään täyttö ennen keräämistä. [24] Mikäli näin ei ole, voi kerääjä joutua odottamaan täytön suorittamista. Tällöin yhden aktiivikeräystehtävän suorittamisessa säästynyt aika s reservikeräykseen verrattuna ei ole vakio. Motuksen kanssa tämä oletus toteutuu, sillä se ei vapauta tilausta keräilyyn, ennen kuin täyttö on suoritettu. Vastaavasti yli nimikkeelle aktiivista varatun tilan kokoiset tilaukset voidaan ohjata keräyssääntöjen avulla kerättäväksi reservialueelta.
- Käsiteltävät nimikkeet ovat fyysisiltä mitoiltaan pieniä. [22] Tämä oletus tulee siitä, että malli antaa tilavuutena nimikkeelle varatun tilan aktiivialueelta. Käytännössä tilavuus pitää muuttaa nimikemääräksi ja sitä pienempi virhe on, mitä pienempiä nimikkeet ovat. Oletuksen toteutuminen riippuu puhtaasti siitä, mitä nimikkeitä tarkasteltavalla varastolla on.
- Aktiivialueen tilavuus on tiedossa. [22] Malli jakaa koko tilavuudesta siivuja tarkastelluille nimikkeille, joten jos tilavuus on väärin, ohjaa malli aktiivipaikoille joko liian paljon tai liian vähän nimikkeitä. Käytännössä tilavuus pystytään laskemaan riittävällä tarkkuudella, kun hyllyjen määrä ja yhden hyllyn mitat ovat tiedossa.
- Nimikkeen kysyntä f_i tarkasteluvälillä on vakio. [22] Käytännössä tämä oletus ei koskaan toteudu, koska kysyntä ei ole koskaan entuudestaan tiedossa. Mallin hyvyys riippukin pitkälti siitä, miten hyvin kysyntä pystytään arvioimaan. Useimmiten ennustetta ei ole saatavilla, jolloin tulevaa menekkiä joudutaan arvioidaan esimerkiksi menneen kysynnän pohjalta. Tällöin ohjaus on enemmänkin reaktiivista kuin proaktiivista.

Lavapohjaisella mallilla on seuraavat oletukset.

- Täyttö pystytään tekemään yksikön siirtona. [4] Tällöin täyttöihin kuluva työ-määrä on suoraan verrannollinen aktiivin kautta kulkevien yksiköiden määrään ja täyttöjen määrä pystytään estimoimaan keruutöiden määränä, joissa kerättävä määrä on vähemmän kuin täyden yksikön määrä. Lisäksi oletus takaa, että täyttöihin keskimäärin kuluva aika c_r pystytään estimoimaan. Käytännössä tämä oletus ei toteutudu, jos monessa yksikössä on useampaa kuin yhtä nimikettä.
- Jokaisen nimikkeen alle täyden yksikön määrät kerätään joko aktiivista tai

reservistä. [4] Täydet yksikkömäärät on järkevää kerätä aina reservialueelta. Mikäli keräys tehtäisiin aktiivialueelta, jouduttaisiin samantien tekemään täyttö, sillä keräys tyhjentää yksikön. Alle täyden yksikön määrät taas on tehokkaampaa kerätä aktiivista, mikäli nimikettä siellä varastoidaan. Tämä oletus pystytään täyttämään Motuksessa keräyssääntöjen avulla.

- Aktiivialueelle mahtuvien yksiköiden maksimimäärä on tiedossa. [4] Malli jakaa aktiivialueen varastopaikat nimikkeille. Jos paikkojen määrä ei ole tiedossa, ohjaa malli aktiivipaikoille joko liian paljon tai liian vähän nimikkeitä. Käytännössä tämä ei ole aina tarkasti tiedossa. Joissain varastoissa käytetään useamman kuin yhden kokoisia yksiköitä, jolloin aktiivialueelle mahtuvien yksiköiden määrä on riippuvainen käytettävistä yksiköistä. Esimerkiksi hyllyväliin voi mahtua neljä EUR-lavaa ja kolme FIN-lavaa, jolloin paikkoja on joko kolme tai neljä riippuen mitä yksiköitä on käytössä. Käytettävien varastopaikkojen määrä voidaan kuitenkin näissäkin tapauksissa estimoida esimerkiksi käytettävien yksikkötyyppien prosenttiosuuksien mukaan.
- Vastaavasti kuin nestemallissakin, olettaa lavapohjainen malli nimikkeen kysynnän olevan vakio tarkasteluvälillä. [22] Tähän pätee samat pohdinnat, kuin nestemallinkin kohdalla.

8.2 Keruutehtävien ja täyttöjen suoritusajan estimointi

Molemmat mallit tarvitsevat toimiakseen arviot keskimääräisestä yhden keruutehtävän suoritusajasta aktiivialueelta ja reservialueelta, sekä täytön suoritusajasta. Nämä voidaan määrittää esimerkiksi seuraamalla ja mittaamalla työntekijöiden työtä. Helpommin ja tarkemmin se onnistuu kuitenkin Motuksen historiatiedoista. Motukseen tallentuu tietoa kaikista toiminnoista, joista on helppo määrittää erilaisia statistiikkoja. Tämä vaatii kuitenkin, että järjestelmään on ehtinyt kertymään dataa, joten aivan heti käyttönotossa sitä ei voi tehdä. Tapahtumia ei myöskään säilötä määräänsä pidempään, sillä varastohallintajärjestelmä on operatiivinen järjestelmä.

Täytön keskimääräinen suoritus aika saadaan, sillä siirroista tallentuu aloitus- ja valmistumisaika järjestelmään. Näistä voidaan laskea täyttösiirtojen keskimääräinen kesto. Aloitus aikaleima tallentuu, kun siirto aloitetaan, eli kun käyttäjä on varmistanut tulleen oikean yksikön kohdalle. Loppuaika taas tallentuu, kun käyttäjä kuittaa siirtäneensä yksikön uuteen sijantiin. Tällöin täytön kesto koostuu yksikön käsittelystä lähtö- ja kohdepaikalla, sekä kulkumatkasta lähtöpaikasta kohdepaikkaan.

Yhden aktiivi- ja reservikeräilytehtävän keskimääräinen suoritus aika taas saadaan laskettua keräilylistojen perusteella. Keräyslista koostuu keräilytehtävistä, jotka käyttäjä kerää yhdellä keräyslenkillä. Keräyslistan kokonaiskesto saadaan katsomalla ensimmäisen ja viimeisen keräystehtävän kuittauksen välinen aika. Kun tämä jaetaan

keräyslistalla olleiden keräystehtävien määrällä, saadaan yhteen keräystehtävään keskimäärin kulunut aika. Tällöin aika koostuu kulkuajasta varastopaikalta toiselle sekä tuotteiden keräämisestä ja käsittelystä eri varastopaikoilla. Tämä laskentatapa ei huomioi keräyksen kohdeyksikön perustamiseen kuluvaan aikaa eikä kohdeyksikön käsittelyyn kuluvaan aikaa viimeisen keräyskuittauksen jälkeen, mikä on syytä huomioida. Yleensä keräyksen kohdeyksikkö pitää vain siirtää pakkaamoon tai lähettämöön keräyslistan valmistumisen jälkeen, mutta tuotteet saatetaan joutua myös jakelemaan useaan paikkaan. Esimerkiksi on mahdollista, että reservikeräilylenkin jälkeen keräilyn pohjalava vain jätetään pakkaamoon, kun taas aktiivikeräilyn pohjalava pitää tyhjentää, jolloin työmäärä on suurempi.

8.3 Kysynnän estimoiminen historiatiedoista

Molemmat mallit tarvitsevat syötteenään nimikkeiden menekin tarkastelujaksolla. Kysyntäennustetta ei yleensä ole saatavilla varastonohjauksen tarpeisiin. Kysynnän ennustaminen on hyvin laaja aihe, josta on tehty useita tutkimuksia ja väitöskirjoja, joten siihen ei perehdytä tässä työssä tarkemmin. Yksinkertainen tapa, jolla voidaan saada edes jonkinlainen estimaatti nimikkeen menekistä, on olettaa menekin olevan sama kuin sen historiassa toteutunut menekki.

Kuten kappaleessa 3 on kuvattu, saa WMS tiedot asiakkaan tilauksista tilausriveineen liiketoimintaprosessista vastaavasta järjestelmästä, joka useinmiten on toiminnanohjausjärjestelmä, eli ERP. WMS luo tilausten pohjalta keräystehtävät, jotka jakaantuvat esimerkiksi aktiivi- ja reservikeräilytehtäviksi kulloisen varastointitilanteen perusteella. Näiden pohjalta WMS ohjaa keräilyä siten, että oikea määrä tuotetta toimitetaan oikealle vastaanottajalle. Varastohallintajärjestelmällä on siis tiedossa milloin ja miten paljon mitäkin nimikettä on tilattu ja miten tuotteet on kerätty.

Yksi päätettävä parametri menekin estimoinnissa vanhojen tilausten perusteella on kuinka pitkältä ajalta menneisyydestä tilauksia tarkastellaan. Mitä lyhyempi väli on, sitä nopeammin kysynnän vaihtelut näkyvät ennusteessa ja päinvastoin. Aikaväliä päätettäessä on hyvä miettiä myös varaston nimikkeiden elinkaarta. Jos nimikkeillä on lyhyt elinkaari, kuten esimerkiksi kulutuselektroniikan tukkukaupassa, ei mennyttä kysyntää kannata tarkastella esimerkiksi vuoden ajalta, koska vuoden vanhojen tuotteiden menekki on helposti jo vähentynyt. Vastaavasti varaosavarastossa nimikkeiden elinkaari voi olla hyvinkin pitkä.

Yksi yleinen ilmiö, jota yksinkertainen historiatietoihin perustuva kysyntämallilla ei kykene ottamaan huomioon, on kausivaihtelut. Tietyillä tuotteilla kysyntä on selvästi kausiluonteista. Esimerkiksi lämmityslaitteiden kysyntä kasvaa merkittävästi talven tullessa ja vastaavasti jäähdytyslaitteiden kesän tullessa. Jos mennyttä kysyntää tarkastellaan liian pitkältä aikaväliltä, ei sesongin alkuun reagoida riittävän

nopeasti.

Nestemallin ja lavapohjaisen mallin käyttämät kysyntäennusteet eroavat hieman toisistaan, minkä vuoksi niitä on tarkasteltu erikseen.

8.3.1 Kysynnän estimoiminen nestemallia varten

Nestemallissa täytyy arvioida tilavuus, joka nimikettä virtaa aktiivialueen läpi tarkasteluvälillä. Tämä voidaan määrittää joko asiakastilausten tai toteutuneiden keräystehtävien pohjalta. Mikäli määrittäminen tehdään tilattujen määrien pohjalta, pitää tästä suodattaa pois isot tilausmäärät, mikäli ne kerätään esimerkiksi suoraan reservialueelta. Nimikkeiden kappalemäärät pystytään helposti muuttamaan tilavuudeksi, kun varastonhallintajärjestelmään on syötetty eri pakkauskokojen mittatiedot.

8.3.2 Kysynnän estimoiminen lavapohjaista mallia varten

Malli tarvitsee syötteenään nimikekohtaisesti alle täyden yksikkömäärän keruutöiden määrän p_i , näillä keruilla tyhjenevien yksiköiden määrän d_i sekä kokolavakeruiden määrän D_i . Nämä voidaan määrittää historiatiedoista kahdella eri tavalla. Ensimmäinen tapa on määrittää jokaiselle nimikkeelle paljonko sitä tyypillisesti vastaanotetaan per lava. Tämän tiedon avulla voi laskea tilatuista määristä paljonko niistä syntyy kokolavakeräyksiä ja paljonko alle täyden lavan keräystehtäviä. Esimerkiksi jos nimikettä x on keskimäärin 20 kpl per lava ja eräässä tilauksessa sitä on tilattu 25 kpl, syntyy tästä keskimäärin yksi täyslavakeruutehtävä ja yksi normaalikeruutehtävä. Tällöin on helppo määrittää myös alle täyden lavan keruutehtävillä tyhjenevien yksiköiden määrä.

Toinen tapa määrittää täyslavakeräysten ja alle täyden lavan keräysten määrä on toteutuneista keräystapahtumista. Tällä tavalla hankaluudeksi tulee määrittää montako lavaa alle täyden lavan keräystehtävillä tyhjenee, sillä tästä ei jää suoraa merkintää Motukseen. Kokoamalla yhteen erilaisia tietoja tämä kuitenkin onnistuu, mutta ei erityisen helposti.

Ensimmäisellä tavalla lasketut ennusteet voivat poiketa paljonkin todellisuudesta, mikäli nimikkeiden määrä vaihtelee paljon lavoittain. Sitä ei myöskään voi hyödyntää, ellei järjestelmä ole ollut jo käytössä hyvän aikaa, jolloin suurimmalle osalle nimikkeistä on ehtinyt kertymään dataa vastaanotosta. Usein kannattaakin käyttää jälkimmäistä tapaa, mikäli vain se on mahdollista.

8.4 Lavapohjaisen mallin muut parametrit

Lavapohjainen malli tarvitsee aiemmin esitettyjen parametrien lisäksi vielä aktiivipaikkojen minimi- ja maksimimäärän l_i ja u_i per nimike. Minimimäärä l_i on usein

järkevä olla yksi, sillä kuten teoreemassa 6.1.1 on esitetty, useamman lavan varastointi ei tuo lisäsäästöjä, ellei nimikkeen koko saldo varastoida aktiiviin. [4] Minimimäärä voi myös olla suurempi kuin yksi, mikäli nimikettä halutaan pitää aktiivialueella esimerkiksi ruuhkautumisen estämiseksi useampi lava. Useamman kuin yhden lavan varastoiminen aktiivialueelle per nimike voidaan perustella myös, mikäli esimerkiksi päivän tai vuoron kysyntä on tarkasti tiedossa ja aktiivialue halutaan täyttää tällä perusteella. Useamman kuin yhden lavan määrittäminen Motuksen nykyisillä täyttösäännöillä ei kuitenkaan onnistu erityisen tarkasti.

Useamman kuin minimimäärän varastoiminen aktiiviin ei tuo lisäsäästöjä, ellei nimikkeen kaikkia yksiköitä varastoida aktiivialueelle. Lavojen maksimimäärän u_i määrittäminen onkin hankalaa, sillä tällöin tulee tuntea nimikkeen lavojen lukumäärän yläraja tarkasteluvälillä. Vaikka koko logistiikkaketjusta on yhä enemmän ja enemmän tietoa saatavilla, on päivän aikana saapuvien kuormien sisältö harvemmin etukäteen tiedossa riittävän tarkalla tasolla. Mikäli tarkempaa tietoa yksiköiden ylärajasta ei ole, voidaan maksimimäärä pitää yhtenä. Tällöin mitään nimikettä ei varastoida aktiiviin yhtä lavaa enempää.

8.5 Mallien tulosten muuntaminen täyttösäännöiksi

Sekä nestemalli että lavapohjainen malli antavat nimikekohtaisesti määrän, jossa sitä kannattaa varastoida aktiivialueelle. Nestemalli antaa määrään tilavuutena ja lavapohjainen malli lavoina. Nämä määrät täytyy muuntaa kappaleessa 7.3 esitetyiksi täyttösäännöiksi, joilla Motus ohjaa täyttöjä.

Nestemallin antama tilavuus, joka varataan nimikkeelle, tulee muuntaa ensin nimikkeen kappalemääräksi. Muunnoksessa voi hyödyntää jotakin lähteessä [21] esitettyä korjausmetodia, jolla jatkuvan mallin tulos saadaan muutettua diskreetiksi. Yksinkertaisin tapa on pyöristää tulos alaspäin lähimpään kokonaislukuun. Tämä on maksimimäärä, jota nimikettä pyritään pitämään aktiivialueella. Täyttösäännöissä pitää aina määrittää hälytysraja, jonka alitus laukaasee täytön. Hälytysrajan voi valita tilanteen mukaan. Esimerkiksi se voi olla prosentuaalinen osuus koko määrästä. Jotta Motus ei normaalitilanteessa ohjaa aktiivialueelle enemmän kuin maksimimäärän nimikkeitä, tulee täyttösäännölle määritettävä minimiraja olla nimikkeen aktiivialueen maksimimäärän ja hälytysrajan erotus. Tämä ei kuitenkaan takaa, etteikö nimikkeen määrä aktiivialueella hetkellisesti ylittäisi sen maksimimäärää, sillä kuten aiemmin on esitetty, Motus tarkastelee aina vapaata saldoa, eikä aktiivialueella olevaa määrää.

Lavapohjainen mallin antama lavojen määrän muuttaminen täyttösäännöiksi onnistuu vain, jos varastoitavien lavojen määrä on yksi tai ääretön. Mikäli määrä on yksi lava, asetetaan täyttösäännön minimirajaksi yksi ja maksimirajaksi ääretön. Tällöin täyttö luodaan aina täsmälleen yhdelle lavalle. Kun nimikkeen kaikki lavat

halutaan varastoida aktiivissa, asetetaan sekä minimi- että maksimiraja äärettömäksi. Tällöin täyttö luodaan kaikille nimikettä sisältäville lavoille. Hälytysrajan voi taas määrittää haluamalla tavalla. Se voi olla esimerkiksi tietty prosentuaalinen osuus keskimääräisen täyden lavan määrästä.

8.6 Saldon huomioiminen

Molemmissa malleissa pitää miettiä, mitkä nimikkeet otetaan laskentaan mukaan. Mikäli mukaan otetaan vain nimikkeet, joilla on laskentahetkellä saldoa, ei mukaan tule nimikkeitä joilla on suuri menekki, mutta jotka ovat hetkellisesti loppuneet. Tällöin kun nimikettä saapuu varastolle, kerätään sitä vain reservistä, mikä on hidasta. Toisaalta taas, jos mukaan otetaan kaikki nimikkeet joita on toimitettu tarkasteltavalla ajanjaksolla, voi nimike varata tilaa aktiivialueelta, vaikka sitä ei olisi pitkään aikaan saldoilla.

8.7 Varastointisääntöjen automaattinen määrittäminen

Mikäli varastointisäännötkin määritetään aktiivalueelle varastoitaville nimikkeille, voidaan nimikettä siirtää vastaanotosta suoraan aktiivialueelle ilman, että tuotteet kiertävät reservialueen kautta. Sekä nestemalli että lavapohjainen malli antavat nimikekohtaisesti määrän, joka kannattaa varastoida aktiivialueelle. Nestemalli antaa määrään tilavuutena ja lavapohjainen malli lavoina. Nestemallin pohjalta on helppo määrittää myös varastointisääntö, sillä varastointisäännölle määritetään maksimiraja kohdealueella kappaleina. Lavapohjaisessa mallissa tämä maksimimäärä joudutaan arvioimaan esimerkiksi keskimääräisen täyden lavamäärän avulla. Varastointisäännölle kannattaisi kuitenkin lisätä myös maksimilavojen määrä.

8.8 Menetelmä hyllypohjaiselle aktiivialueelle

Aiemmissa kappaleissa esitetyn analyysin pohjalta kehitettiin nestemallia hyödyntävä menetelmä täyttösääntöjen automaattiseksi määrittämiseksi hyllypohjaisella aktiivialueella. Se on esitetty algoritmissa 5. Menetelmä valitsee aktiivialueelle varastoitavat nimikkeet sekä määrät ja luo niiden pohjalta valituille nimikkeille täyttösäännöt. Täyttöön keskimäärin kuluva aika c_r ja säästö s , kun kerätään aktiivialueelta reservialueen sijaan, määritetään osiossa 8.2 esitetyllä tavalla. Menekki estimoidaan menneistä toteutuneista keräystehtävistä. Joukko P_i sisältää nimikkeen i kerätyt määrät tarkasteluvälillä. Menetelmä hyödyntää nestemallin heuristista algoritmia 2 nimikkeiden ja niiden tilavuuksien laskentaa varten.

Algoritmi 5: Nestemallia hyödyntävä menetelmä täyttösääntöjen luomiseksi nimikkeille hyllypohjaisella aktiivialueella

Syötteet : Nimikkeiden kerätyt määrät tarkasteluvälillä P_i
 Nimikkeiden pituus, leveys ja korkeus w_i, l_i, h_i
 Säästö keskimäärin kun kerätään aktiivista eikä reservistä s
 Täyttöön keskimäärin kuluva aika c_r
 Aktiivialueen tilavuus V

Tulokset : Täyttösäännöt nimikkeittäin

```

1 // Määritetään nimikkeen toimitettu tilavuus  $f_i$  ja
2 // keräystehtävien määrä  $p_i$ 
3 for  $i = 1, 2, \dots, N$  do
4   foreach  $p \in P_i$  do
5      $f_i := f_i + o * w_i * l_i * h_i$ 
6      $p_i := p_i + 1$ 
7   end
8 end
9 Määritä algoritmilla 2 nimikkeittäin aktiivialueelle varastoitavat nimikkeet  $S$  ja
  määrät  $v_i^*$ 
10 foreach  $i \in S$  do
11   // Muutetaan tilavuus  $v_i^*$  määräksi  $qty_i$ 
12    $qty_i := v_i^* / (w_i * l_i * h_i)$ 
13   Luo täyttösääntö nimikkeelle  $i$ . Minimiraja := 1, maksimiraja :=  $qty_i$ 
14 end

```

8.9 Menetelmä lavapohjaiselle aktiivialueelle

Aiemmissa kappaleissa esitetyn analyysin pohjalta kehitettiin lavapohjaiseen malliin perustuva menetelmä täyttösääntöjen automaattiseksi määrittämiseksi lavapohjaisella aktiivialueella. Se on esitetty algoritmissa 6. Menetelmä antaa jokaiselle aktiivialueelle varastoitavalle nimikkeelle täsmälleen yhden varastopaikan, sillä Motuksen täyttösäännöillä ei pystytä luotettavasti pitämään tiettyä määrää lavoja aktiivialueella. Täyttöön keskimäärin kuluva aika c_r ja säästö s , kun kerätään aktiivialueelta reservialueen sijaan, määritetään osiossa 8.2 esitetyllä tavalla. Kysyntä estimoidaan nimikkeiden menneistä toteutuneista tilauksista. Joukko O_i sisältää nimikkeen i tilatut määrät tarkasteluvälillä. Tähän päädyttiin, koska tyhjenevien lavojen määrä on hankala määrittää keräystapahtumien pohjalta. Tilatuista määristä lasketaan mallin heuristista algoritmia 3 varten alle lavamäärän keruutehtävien määrä p_i , tyhjenevien lavojen määrä d_i ja kokolavakeruiden määrä D_i . Tämän jälkeen aktiivialueelle varastoitaville nimikkeille luodaan täyttösäännöt.

Algoritmi 6: Lavapohjaista mallia hyödyntävä menetelmä täyttösääntöjen luomiseksi lavapohjaisella aktiivialueella

Syötteet : Nimikkeittäin keskimääräinen määrä nimikettä lavalla m_i
 Nimikkeiden tilatut määrät tarkasteluvälillä O_i
 Säästö keskimäärin kun kerätään aktiivista eikä reservistä s
 Täyttöön keskimäärin kuluva aika c_r
 Aktiivipaikkojen määrä V

Tulokset : Täyttösäännöt nimikkeittäin

```

1 // Määritetään parametrit  $p_i$ ,  $d_i$  ja  $D_i$  algoritmia 3 varten
2 for  $i = 1, 2, \dots, N$  do
3   // Muodostetaan nimikkeelle  $i$  tilattujen määrien  $O_i$  pohjalta alle
4   // lavamäärien keruutehtävien määrä  $p_i$ , tyhjenevien lavojen
5   // määrä  $d_i$  ja kokolavojen määrä  $D_i$ 
6   foreach  $o \in O_i$  do
7     if  $o \leq m_i$  then
8        $p_i := p_i + 1$ 
9        $d_i := d_i + o/m_i$ 
10    else
11       $D_i := D_i + \text{Floor}(o/m_i)$ 
12      if  $\text{Mod}(o, m_i) \neq 0$  then
13         $p_i := p_i + 1$ 
14         $d_i := d_i + \text{Mod}(o, m_i) / m_i$ 
15      end
16    end
17  end
18 end
19 Aseta kaikille nimikkeille  $l_i := 1$  ja  $u_i := 1$ 
20 Määritä algoritmilla 3 nimikkeittäin aktiivialueelle varastoitava lavamäärä  $x_i$ 
21 for  $i = 1, 2, \dots, N$  do
22   if  $x_i > 0$  then
23     Luo täyttösääntö nimikkeelle  $i$ . Minimiraja := 1, maksimiraja := ääretön
24   end
25 end

```

9. AUTOMAATTISTEN TÄYTTÖSÄÄNTÖJEN VERTAILU SIMULOINNILLA

Motuksen ohjaamista manuaalivarastoista tyypillisin aktiivialue on lavavaraston lattiapaikoista ja kuormalavahyllyjen alimmista hyllyistä koostuva aktiivialue, joihin varastointi tehdään lavoina. Lavapohjainen malli soveltuu juuri tällaisen aktiivialueen ohjaamiseen. Lavapohjaisen mallin pohjalta kappaleessa 8 kehitetyn algoritmin 6 toimivuutta täyttösääntöjen automaattiseen muodostamiseen haluttiin testata olemassa olevassa prosessissa ja verrata Motuksen nykyisin käytössä olevaan algoritmiin 4. Vertailu päätettiin toteuttaa simuloimalla. Tavoitteena oli tarkastella, miten hyvin lavapohjaisen malliin perustuvan menetelmän avulla luodut täyttösäännöt pystyvät ohjaamaan aktiivialuetta. Tarkemmin kiinnosti, tehostuuko kokonaisprosessi lavapohjaisen mallin pohjalta luoduilla täyttösäännöillä. Vertailukohtana käytettiin nykyisin käytössä olevaa täyttösääntöjen automaattista muodostusta, joka on kuvattu kappaleessa 7.4.1 ja algoritmissa 4.

Vertailukohteeksi pohdittiin kahta Motus-asennusta. Molemmat asiakkaat X ja Y ovat vähittäiskaupan manuaalivarastoja, joissa lavavarasto on jaettu aktiivi- ja reservialueeseen. Asiakkaalla Y on kuitenkin huomattavan paljon sekalavoja, jotka sisältävät useampaa kuin yhtä nimikettä. Näin ollen täytöt reservistä aktiiviin eivät usein ole lavasiirtoja, vaan tuotteet joudutaan keräämään usealta lavalta. Täten lavapohjaisen mallin oletus täyttöjen toteutumisesta lavasiirtoina ei toteudu, eikä malli välttämättä sovellu kyseisen varaston aktiivialueen ohjaamiseen. Asiakkaan X varastossa ei pääasiassa ole sekalavoja, joten täytöt pystytään tekemään lavasiirtoina ja malli soveltuu prosessiin hyvin. Täten asiakkaan X prosessi valittiin simuloinnin kohteeksi.

9.1 Simuloitava prosessi

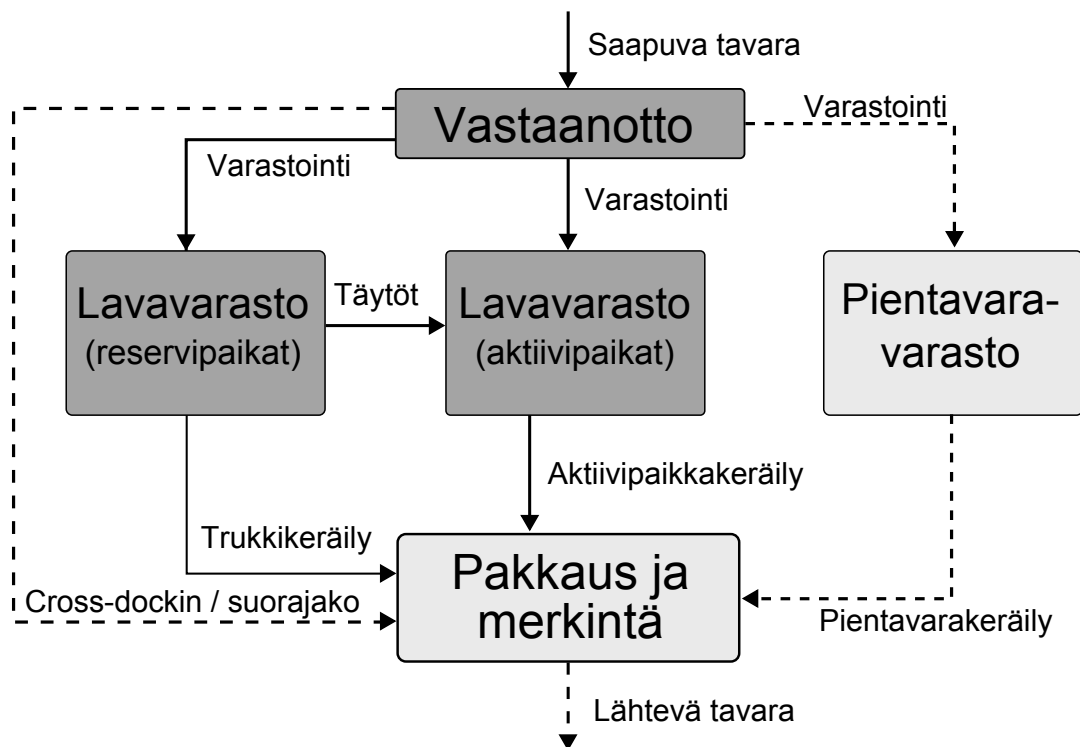
Simuloitavaksi prosessiksi valittiin asiakkaan X prosessi. Asiakas on Suomessa toimiva vähittäiskauppa, joka toimittaa konsernin myymälöihin tavaraa logistiikkakeskuksesta. Kyseessä on manuaalivarasto, jonka materiaalivirrat on esitetty yksinkertaistettuna kuvassa 9.1. Koska simuloinnin tavoitteena on tarkastella nimenomaan aktiivi- ja reservialueen ohjausta, jätettiin muut prosessit, kuten pientavaravarasto ja cross-docking tarkastelun ulkopuolelle.

Saapuva tavara tunnistetaan ja yksiköidään vastaanotossa, mistä se varastoidaan

lavavarastoon. Asiakkaalla X ei ole käytössä varastointisääntöjä, vaan Motus ohjaa kaikki lavat reserviin. Laskettu varastopaikka reservialueelta on kuitenkin vain ehdotus, jolloin käyttäjä voi varastoida lavan myös suoraan aktiivialueelle. Tällä hetkellä varastointi menee pitkälti käyttäjien oman päätöksen mukaan. [31]

Lavavarasto on jaettu kahteen keräilyalueeseen: reservipaikoilta tehdään trukkeräilyä ja aktiivipaikoilta aktiivipaikkakeräily. Trukkeräily on eräänlaista jakelua. Lähettämössä on alue jokaiselle myymälälle, johon toimitetaan tavaraa. Kerääjä noutaa lavan varastosta trukilla ja jakaa lähettämöön jokaisen toimipisteen ruutuun tilatun määrän verran lavalla olevaa tuotetta. Mikäli koko lavan sisältö on varattu samalle toimipisteelle, siirretään lava kokonaisuudessaan lähettämöön, eli kyseessä on täyslavakeräys. Jakelun jälkeen kerääjä palauttaa lavan varastoon. Mikäli lava tyhjenee keräilyn päätteeksi, ei lavaa tietenkään palauteta varastoon. [31]

Aktiivipaikkakeräily tehdään tuotekeräilyinä, eli käyttäjä kerää kaikkia tilauksia samaan aikaan. Keräily suoritetaan matalakeräilijällä ja tuotteet kerätään kuormalavalle yhteen läjään. Käyttäjä kerää lavalta varatun määrän tuotteita ja siirtää ne keräilyn kohdelavalle. Keräilyä jatketaan, kunnes lava täyttyy. Tämän jälkeen kerääjä käy jakelemassa tuotteet lähettämössä sijaitseville myymäläkohtaisille alueille. Varastolla on tarkoitus lähitulevaisuudessa siirtyä tilauskeräilyyn, jossa kerätään muutamaa tilausta samalla aikaa tuotteiden sijaan. Tämän pitäisi nopeuttaa muun muassa tilausten läpimenoaikaa. [31]



Kuva 9.1: Asiakkaan X yksinkertaistetut materiaaliveirrat. Vaaleammalla esitettyjä prosesseja ja katkoviivalla esitettyjä materiaaliveirtoja ei mallinnettu

Aktiivialue koostuu kahdeksan käytävän kuormalavahyllyjen lattiapaikoista, sekä heti niiden yläpuolella olevasta orsipaikasta. Tätä korkeammalla ovat orsipaidat sekä muut käytävät ovat reservialuetta. Yhteensä aktiivipaikkoja on 900 kappaletta. Varastolla on käytössä sekä EUR- että FIN-lavojaja yhteen varastopaikkaan mahtuu joko 4 EUR-lavaa tai 3 FIN-lavaa. Näin ollen aktiivialueelle mahtuu yhteensä 2700-3600 lavaa. Kun paikkaan varastoidaan ensimmäinen lava, ei samaan paikkaan saa laittaa enää muun tyyppisiä lavoja.

Asiakkaalla on käytössä Motuksen nykyinen automaattinen täyttösääntöjen laskenta, joka on kuvattu algoritmissa 4. Korkeimpaan menekkiluokkaan A on määritetty 8 prosenttia saldolla olevista nimikkeistä. Toiseksi korkeimpaan menekkiluokkaan B taas 22 prosenttia saldolla olevista nimikkeistä. Loput nimikkeet asetetaan pienimpään menekkiluokkaan C. Kyseisellä asiakkaalla vain luokan A nimikkeille luodaan automaattinen täyttösääntö. Täyttösääntöihin asetetaan hälytysrajaksi 5 % täyden yksikön määrästä ja minimitäyttömääräksi 10 % täyden yksikön määrästä. Maksimitäyttöraja on ääretön. Motuksen automaattisen täyttösääntöalgoritmin näkökulmasta tämä tarkoittaa, että vain 8 prosentille nimikkeistä luodaan automaattisesti täyttösääntö. 10 % minimitäyttömäärä tarkoittaa, että käytännössä jokaista nimikettä siirretään aktiivialueelle yksi lava, sillä täyttö tehdään aina kokonaisille lavoille.

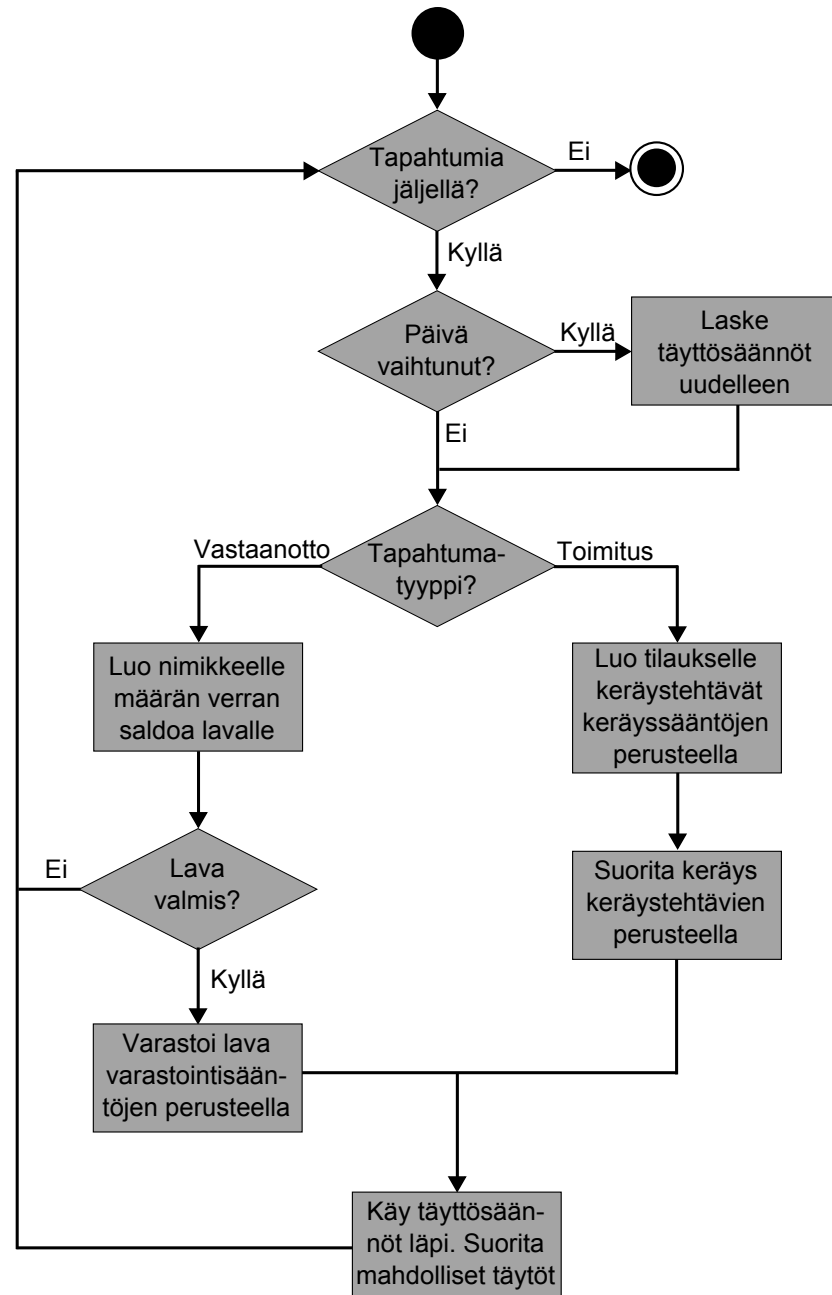
9.2 Simuloinnin toteutus

Simulointia varten toteutettiin oma simulaattori C#-kielellä. Tähän päädyttiin muutamasta syystä. Ensinnäkin sopivan simulaattoriohjelmiston etsiminen ja mallin rakentaminen sillä olisi vaatinut paljon opettelua ja ollut siten työlästä. Toiseksi omalla simulaattorilla mallin tekemä ohjaus saatiin varmasti vastaamaan Motuksen toimintaa ja kolmanneksi simulaattorista saatiin juuri sellaiset statistiikat kerättyä kuin haluttiin. Simulaation oikea toiminta varmistettiin ajamalla simulaattoria askel kerrallaan sekä tarkastelemalla tuloksia kriittisesti.

Simuloinnin tavoitteena oli kokeilla, miten eri tavat luoda täyttösäännöt automaattisesti vaikuttavat keräilytöiden jakaantumiseen aktiivi- ja reservialueelle. Lisäksi tarkasteltiin, miten paljon täyttöjä syntyy aktiivialueen ylläpitämisestä. Simulointi päätettiin toteuttaa diskreetteinä tapahtumina, sillä siten se pystyttiin pitämään yksinkertaisena, mutta kuitenkin riittävällä tarkkuudella tarkasteltavaa prosessia kuvaavana.

Simuloinnissa hyödynnettiin asiakkaan X prosessista kertynyttä dataa. Varaston saldotiedot otettiin talteen tietyltä ajanhetkeltä, jota käytettiin kaikissa simulaatioissa lähtöhetkenä. Myös vastaanoton historiatiedot siirrettiin simulaatiota varten. Täten saapuva tavaravirta saatiin vastaamaan täsmälleen oikeasti toteutunutta. Vastaavasti kaikki myyntitilaukset otettiin talteen niiden luontiaikoineen. Täten myös lähtevä tavaravirta vastaa pitkälti oikeasti toteutunutta.

Simulaatio toimii kuten muutkin tapahtumapohjaiseen malliin perustuvat simulaatiot. Simulaation toiminta on esitetty kuvassa 9.2. Simulaatio ajettiin aina tietyllä aikavälillä, esimerkiksi kolmen kuukauden ajalle. Kyseiseltä aikaväliltä ladattiin kaikki tapahtumat ja ne järjestettiin nousevaan järjestykseen niiden aikaleiman mukaan. Aloitusaajanhetkeksi asetettiin ensimmäisen tapahtuman aikaleima. Kun tämä tapahtuma oli suoritettu, siirryttiin seuraavaan tapahtumaan ja tätä jatkettiin, kunnes kaikki tapahtumat oli käsitelty.



Kuva 9.2: Simulaattorin toimintakaavio.

Tapahtumia oli kahdenlaisia, vastaanotto- sekä toimitustapahtumia. Vastaanotto-tapahtuma koostui nimikkeestä, määrästä ja yksiköstä. Yksikköön luotiin nimikkeelle

saldoa määrän verran. Mikäli tapahtuma oli viimeinen vastaanottotapahtuma kyseiselle yksikölle, varastoitii yksikkö nimikkeen varastointisääntöjen perusteella. Varastoinnin jälkeen käytiin nimikkeen täyttösäännöt läpi ja niiden pohjalta luotiin täyttö, mikäli hälytysraja oli alitettu.

Toimitustapahtuma koostui yhdestä myyntitilauksesta riveineen, eli nimikkeistä ja tilatuista määristä. Myyntitilaukselle luotiin keräyssääntöjen pohjalta keräystehtävät, jotka suoritettiin samantien. Keräyssääntöjä oli kolme. Ensin tarkasteltiin, pystyykö tilatun määrän tekemään täyslavakeräilynä reservialueelta. Tämän jälkeen katsottiin, pystyykö keräyksen tekemään aktiivialueelta. Viimeisenä tarkasteltiin reservialue. Keräilyn yhteydessä tyhjentyneet yksiköt poistettiin, jolloin ne vapauttivat varastopaikan. Toimitustapahtuman lopuksi, kun keräystehtävät oli muodostettu ja suoritettu, käytiin kerättyjen nimikkeiden täyttösäännöt läpi. Mikäli tästä syntyi täyttöjä, suoritettiin ne samantien.

Todellisuudessa myyntitilauksista tehdään ensin saldovaraus, minkä jälkeen niistä muodostetaan keräystehtävät, kun käyttäjä aloittaa keräilyn. Tällä mahdollistetaan se, että useampi saldovaraus voi yhdistyä yhdeksi keräystehtäväksi. Jos esimerkiksi kolme toimipistettä on tilannut samaa tuotetta, syntyisi tästä aktiivialueella yksi keräilytehtävä kolmen sijaan, jos saldovaraus kohdistuu saman yksikön saldoon. Simulaation antama keräystehtävien määrä, ja siten myös kokonaistyömäärä, on siis aktiivialueella todellisuutta suurempi. Mikäli tämä yhdistyminen olisi otettu simulaatiossa huomioon, olisi simulaation pitänyt ottaa huomioon mm. keräystä tekevien työntekijöiden määrä sekä keräyksen jälkeen jakeluun menevä aika, mikä olisi monimutkaistanut simulaatiota huomattavasti. Tämä on kuitenkin hyvä pitää tiedostaa tuloksia tarkasteltaessa.

Kun yhden päivän kaikki tapahtumat oli käsitelty, suoritettiin täyttösääntöjen automaattinen laskenta. Täyttösäännöt luotiin siis aina jokaiselle päivälle uudestaan. Simuloinneilla verrattiin Motuksessa tällä hetkellä käytössä olevaa täyttösääntöjen laskenta-algoritmia 4, sekä tässä työssä kehitettyä, lavapohjaiseen malliin pohjautuvaa täyttösääntöjen laskenta-algoritmia 6. Lavapohjaisen mallin algoritmille annettiin aktiivialueen varastopaikkojen määräksi 3200. Koska asiakkaan X asetuksilla nykyisin käytössä oleva automaattinen laskenta luo täyttösäännöt vain kahdeksalle prosentille nimikkeistä, vertailtiin reiluuden vuoksi samaa algoritmia myös siten, että täyttösäännöt luodaan samalle määrälle nimikkeitä.

Historiatiedoista määritettiin yhden täytön keskimääräiseksi suoritusajaksi c_r 178 sekuntia. Vastaavasti yhden aktiivikeruutehtävän keskimääräiseksi suoritusajaksi määritettiin c_1 167 sekuntia ja reservikeruutehtävän c_2 231 sekuntia, jolloin säästöksi s saatiin 64 s. Keräystehtävien keskimääräinen suoritus aika laskettiin kahden peräkkäisen keräystapahtuman keskiarvona. Siinä ei siis ole huomioitu aktiivikeräilyn jälkeen tehtävää jakelua. Kysynnän estimointia varten huomioitiin edellisen 30

päivän aikaiset tilaukset. 30 päivää valittiin siksi, että myös ABC-luokittelu tehdään edellisen 30 päivän ottokertojen perusteella.

Simuloinnin yhteydessä kerättiin erilaisia statistiikkoja. Oleellimmat ovat täyttöjen kokonaismäärä sekä täyslava-, aktiivi- ja reservikeräilytehtävien kokonaismäärä. Lisäksi aktiivialueen täyttöaste, eli kuinka monta varastopaikkaa on prosentuaalisesti varattuna, tallennettiin jokaiselta simulointipäivältä.

9.3 Simuloinnin tulokset

Simuloinnista saadut tulokset on esitetty taulukossa 9.1. Simuloinnit ajettiin viiden eri aikavälin datalla. Simulaatiot #1-#4 tehtiin kolmen kuukauden vastaanotto- ja tilausdatalla ja simulaatio #5 yhdeksän kuukauden datalla. Algoritmi M8% kuvaa asiakkaalla nykyisin käytössä olevaa automaattista täyttösääntöjen laskenta-algoritmia 4, joka luo säännöt kahdeksalle prosentille nimikkeistä. Algoritmi Y3200 on lavapohjaiseen malliin perustuva täyttösääntöjen luontialgoritmi 6, jossa aktiivialueen kooksi asetettiin 3200 varastopaikkaa. M3200 on Motuksen nykyinen algoritmi 4, mutta parametrit asetettiin siten, että täyttösääntöjä luotiin saman verran kuin lavapohjaisella menetelmällä. Sarakkeessa \sum on kuvattu kokonoistymäärä, joka on saatu laskemalla yhteen jokaisessa tehtävässä keskimäärin yhteensä kulunut aika (eli aktiivikeruiden määrä on kerrottu yhdessä aktiivikeruussa keskimäärin kuluneella ajalla 167 sekuntia ja niin edespäin). Täyslavakeruun arvioitiin kestävän keskimäärin saman verran kuin täytön, eli 178 sekuntia. Sarakkeessa Δ on kuvattu miten paljon algoritmin kokonaistymäärä \sum eroaa algoritmin Y3200 kokonaistymäärästä.

Tuloksista voidaan nähdä, että algoritmi M8% pärjää selvästi heikoiten, kuten oletettaa saattaakin. M8% luo täyttösääntöjä huomattavasti vähemmän kuin mitä aktiivipaikkoja on. Tästä syystä täyttöjä ei synny riittävästi, minkä vuoksi aktiivialueen keskimääräinen täyttöaste jää matalaksi ja aktiivikeruiden määrä jää huomattavasti pienemmäksi kuin muilla algoritmeilla.

Mieleenkiintoisempaa on algoritmien M3200 ja Y3200 vertaaminen, joilla on samat lähtökohdat. Molemmat jakavat saman verran varastopaikkoja nimikkeille aktiivialueelta ja niillä on käytössä sama kysyntäennuste, joten algoritmien suoriutuminen määräytyy puhtaasti sen mukaan, miten hyvin ne valitsevat aktiivialueelle varastoitavat nimikkeet. Tuloksista nähdään, että Y3200 pärjää hieman paremmin kaikissa testitapauksissa. Algoritmillä aktiivialueen täyttöaste on korkeampi ja siten myös aktiivikeruiden määrä. Voidaan siis sanoa, että Y3200 kykenee ohjaamaan aktiivialuetta tehokkaammin.

Erityisen huomionarvoista on, että algoritmilla Y3200 syntyy useassa simulaatiossa vähemmän täyttöjä kuin algoritmilla M3200. Algoritmi kykenee siis tehokkaampaan aktiivialueen ohjaamiseen pienemmällä työmäärällä sen ylläpitämiseksi. Tämä on myös oletettavaa, sillä M3200 valitsee nimikkeet puhtaasti menekin perusteella, kun

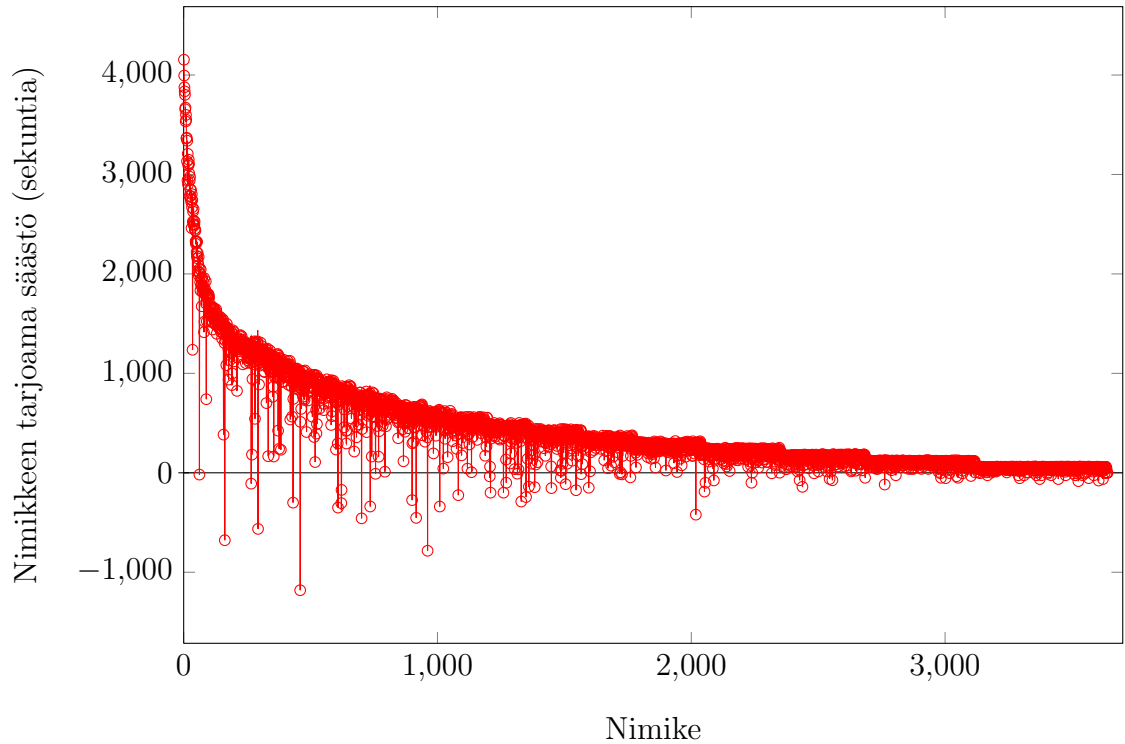
Taulukko 9.1: Simuloinnin tulokset ilman varastointisääntöjä. Σ on kokonaistyömäärä sekunneissa, joka on laskettu kaavalla $ac_1 + rc_2 + kc_3 + tc_r$, missä a on aktiivikeruiden määrä, c_1 aktiivikeruussa kuluva aika, r reservikeräysten määrä, c_2 reservikeruussa kuluva aika, k täyslavakeruiden määrä, c_3 täyslavakeruussa kuluva aika, t täyttöjen määrä ja c_r täyttöön kuluva aika. Δ on prosentuaalinen ero kokonaistyömäärässä Y3200 algoritmiin verrattuna.

	Alg.	Akt.-keruut	Res.-keruut	Täysl.-keruut	Täytöt	Σ	Δ	Keskim. täyttöaste
#1	M8%	35106	13785	2326	807	9604711	3,53 %	84,70 %
	M3200	41392	7834	1905	1516	9331056	0,58 %	89,77 %
	Y3200	43430	5738	1961	1967	9277472	-	94,02 %
#2	M8%	37359	16511	2000	982	10583790	3,98 %	88,17 %
	M3200	44834	9273	1636	1748	10231693	0,52 %	94,9 %
	Y3200	45834	8175	1760	1815	10179053	-	97,18 %
#3	M8%	45665	21187	2501	1156	13171198	5,38 %	82,68 %
	M3200	57874	9491	1876	2314	12603199	0,83 %	92,18 %
	Y3200	59016	8230	2023	2148	12499240	-	93,40 %
#4	M8%	47953	25673	2489	1368	14625160	4,95 %	82,17 %
	M3200	63069	11094	1770	3738	14075661	1,00 %	94,45 %
	Y3200	63073	10925	1996	2939	13935296	-	94,02 %
#5	M8%	114046	82059	6936	4424	40023391	7,43 %	78,30 %
	M3200	159358	37724	4996	6978	37458402	0,55 %	87,95 %
	Y3200	161842	34958	5498	6583	37253330	-	89,33 %

taas Y3200 ottaa huomioon myös täyttöihin kuluvaan työmäärän. M3200 valitsee aktiivialueelle siis nopeimmin kiertävät nimikkeet, mutta tiettyjen nimikkeiden pitämiseen aktiivialueella kuluu enemmän täyttöjä. Tätä on havainnollistettu kuvassa 9.3, jossa on esitetty kaavalla 6.10 laskettu nimikekohtainen säästö, jos nimikettä varastoidaan tasan yksi yksikkö aktiivialueelle. Laskenta tehtiin simulaation lähtödatalla. Nimikkeet on järjestetty niiden ottokertojen mukaiseen järjestykseen suurimmasta pienimpään. Osalla nimikkeistä säästö on negatiivinen, eli sitä ei kannata varastoida aktiivialueelle vain yhtä lavaa.

Ero kokonaistyömäärässä algoritmien M3200 ja Y3200 välillä ei ole suuri. Ero kuitenkin kasvaa, mitä enemmän säästöä aktiivikeräys tuo reservikeräilyyn nähden. Pienemmän työmäärän lisäksi Y3200 kykenee parempaan aktiivialueen täyttöasteeseen, jolloin aktiivialue on tehokkaammassa käytössä.

Asiakkaalla X Motus ei ohjaa varastointia, vaan varastointi on käyttäjien vastuulla. Tämän vuoksi taulukon 9.1 tuloksissa kaikki varastointi on tehty reserviin. Suoraan



Kuva 9.3: Nimikkeiden tarjoamat säästöt, jos nimikettä pidetään aktiivialueella. Säästö on laskettu kaavalla 6.10 ja nimikkeet on järjestetty keräystehtävien määrän p_i mukaan suurimmista pienimpään.

aktiivialueelle tehtävän varastoinnin vaikutusta prosessin tehokkuuteen haluttiin kuitenkin myös testata simuloimalla. Näiden simulointien tulokset on esitetty taulukossa 9.2. Kuten odottaa saattaa, vähenee täyttöjen määrä, kun varastointi voidaan tehdä suoraan aktiivialueelle. Täyttöjen määrä vähenee n. 9-20 % taulukon 9.1 tuloksiin verrattuna. Tämä on hyvin linjassa myös Hollingsworthin esittämien tulosten kanssa, jolla suora varastointi aktiivialueelle väheni myös n. 20 % [20]. Yleisesti voidaankin todeta, että varastointisäätöjen asettaminen tehostaa kokonaisprosessia.

Prosesseja tarkasteltaessa on aina hyvä pitää kokonaisuus mielessä. Tehostuuko kokonaisprosessi, jos aktiivikeruiden määrä nousee reservikeräilyn kustannuksella? Toteutetulla simuloinnilla ei pystytä näkemään, mitkä vaikutukset ovat kokonaisprosessiin. Aktiivikeräys on nopeampi tehdä kuin reservikeräily, mutta keräilyn jälkeen tuotteet pitää vielä jakaa lähettämöalueelle asiakaskohtaisesti. Toisaalta myös reservikeräilyssä tuotteet pitää jaella, joten aktiivikeräily on kokonaisuutena todennäköisesti nopeampaa kuin reservikeräily. Varmuudella tätä ei kuitenkaan pysty sanomaan ilman tarkempaa tarkastelua.

Asiakkaalla on tarkoitus siirtyä keräämään tilauksia tuotteiden sijaan. Tämän jälkeen jakelu tehdään jo kerätessä. Keräys tehdään suoraan asiakaskohtaisiin muovilaatikoihin, jolloin jakelu hoituu nopeammin jättämällä muovilaatikko oikeaan

Taulukko 9.2: Simuloinnin tulokset varastointisääntöjen kanssa. \sum on kokonaistyömäärä sekunneissa, joka on laskettu kaavalla $ac_1 + rc_2 + kc_3 + tc_r$, missä a on aktiivikeruiden määrä, c_1 aktiivikeruussa kuluva aika, r reservikeräysten määrä, c_2 reservikeruussa kuluva aika, k täyslavakeruiden määrä, c_3 täyslavakeruussa kuluva aika, t täyttöjen määrä ja c_r täyttöön kuluva aika. Δ on prosentuaalinen ero kokonaistyömäärässä Y3200 algoritmiin verrattuna.

		Akt.- keruut	Res.- keruut	Täysl.- keruut	Täy- töt	\sum	Δ	Keskim. täyttöaste
#1	M3200	39583	9505	2060	1123	9372590	1,47 %	88,14 %
	Y3200	43599	5571	1956	1804	9237214	-	94,24 %
#2	M3200	43385	10578	1749	1397	10248801	1,37 %	93,09 %
	Y3200	46235	7737	1757	1623	10110132	-	97,57 %
#3	M3200	54990	12155	2073	1746	12670917	2,12 %	89,51 %
	Y3200	59550	7676	2022	1853	12407756	-	93,93 %
#4	M3200	59151	14794	2012	2152	14036823	1,25 %	90,50 %
	Y3200	63405	10565	2011	2674	13863080	-	94,61 %
#5	M3200	160140	36710	5079	6131	37218770	1,16 %	87,86 %
	Y3200	165539	31137	5482	5502	36792812	-	90,47 %

ruutuun. Tämä tekee aktiivikeräyksestä kokonaisuutena nopeampaa, jolloin kokonaisprosessi hyötyy, jos keräilytehtäviä saadaan kohdistettua aktiivialueelle enemmän. Tässä tilanteessa uuden menetelmän hyödyntäminen vähentää varmasti työmäärää.

Yleisesti voidaan todeta lavapohjaiseen malliin pohjautuvan algoritmin 6 toimivan erittäin hyvin. Yksinkertaisemmassa prosessissa, jossa keräilyn jälkeen ei tarvitse tehdä erillistä jakelua, vaan jätö tehdään lähettämöön, on aktiivikeräysprosessi huomattavasti reservikeräilyä nopeampaa. Tällöin aktiivikeräilyjen määrä on suoraan verrannollinen prosessin kokonaistehokkuuteen ja algoritmi tarjoaa erinomaisen keinon aktiivialueen automaattiseen ohjaamiseen.

Simulointien tuloksista voidaan yhteenvedona todeta, että tässä työssä esitetty, lavapohjaiselle aktiivialueelle tarkoitettu algoritmi 6 suoriutuu paremmin aktiivialueen ohjauksesta kuin Motuksen nykyinen malli. Ero niiden välillä ei asiakkaan X prosessissa ole suuri, kunhan määrä, jolle täyttösäännöt luodaan, on järkevästi määritetty. Varastoprosessien kehittäminen tapahtuu kuitenkin pienin askelin, joten jokainen parannus on tarpeen. Lisäksi voidaan todeta, että yleisemmässä tapauksessa ero on todennäköisesti suurempi, kuin simuloinneissa on saatu. Tapa, jolla Motuksen nykyisen algoritmin 4 luomien täyttösääntöjen määrä on sidottu ABC-luokkaan, ei ole järkevä, koska sillä ei välttämättä ole mitään tekemistä aktiivialueen koon kanssa. Varastointisäännöt taas kannattaa aina määrittää, koska niiden avulla saadaan ohjausta tehostettua enemmänkin.

10. JOHTOPÄÄTÖKSET

Työn tavoitteena oli kehittää Motus varastonhallintajärjestelmän aktiivialueen ohjausta. Työn pohjalta löydettyjen mallien ja niistä kehitettyjen menetelmien pohjalta voidaan sanoa, että työssä on onnistuttu erinomaisesti.

Aktiivialueet voidaan jakaa niiden luonteen perusteella hyllypohjaisiin ja lavapohjaisiin. Hyllypohjaisella aktiivialueella varastoidaan pieniä tuotteita pahvi- tai muovilaatikoissa esimerkiksi pientavara- tai läpivirtaushyllyissä. Lavapohjaisella aktiivialueella tuotteet varastoidaan lavalla esimerkiksi lattiapaikoille. Molempia aktiivialuetyyppejä on tutkittu kirjallisuudessa.

Aktiivialueesta saatava hyöty määräytyy pitkälti sen perusteella, mitä nimikkeitä aktiivialueella varastoidaan ja miten paljon. Motuksessa ohjaus tehdään täyttösääntöjen perusteella, joilla määritetään mitä nimikkeitä täytetään reservialueelta aktiivialueelle ja miten paljon. Täyttösääntöjen määrittäminen käsin on työlästä eikä sillä useinkaan saada valittua oikeita nimikkeitä, minkä vuoksi valinta halutaan tehdä automaattisesti. Työ osoittaa, että sekä hylly- että lavapohjaiselle aktiivialueelle pystytään määrittämään täyttösäännöt automaattisesti.

Valinta nimikkeistä ja määristä hyllypohjaisella aktiivialueella onnistuu nestemallilla, kunhan varastoitavat nimikkeet ovat mitoiltaan pieniä. Mallin pohjalta kehitettiin menetelmä, jolla pystytään luomaan Motuksen täyttösäännöt automaattisesti. Menetelmää ei kuitenkaan kokeiltu käytännössä tämän työn puitteissa. Menetelmän toimivuutta sekä tehokkuutta olisikin syytä tutkia tarkemmin.

Lavapohjaista aktiivialueetta varten Motuksessa on nykyisin olemassa algoritmi, jolla pystytään luomaan täyttösäännöt automaattisesti. Algoritmille ei kuitenkaan pysty määrittämään aktiivialueen kokoa, minkä vuoksi algoritmi täyttää aktiivialueen vajaasti. Motuksen algoritmi ei myöskään huomioi täyttöihin kuluva työtä, minkä vuoksi aktiivialueelle saatetaan valita väriä nimikkeitä. Kirjallisuudessa esitetyn lavapohjainen mallin pohjalta kehitettiin menetelmä täyttösääntöjen laskemiseksi lavapohjaisella aktiivialueella. Menetelmä ei kuitenkaan sovellu varastoon, jossa on huomattava määrä sekalavoja, koska täyttö pitää pystyä tekemään lavasiirtona. Motusasiakkaan datan pohjalta suoritettujen simulaatioiden perusteella näyttäisi siltä, että menetelmä kykenee ohjaamaan aktiivialuetta Motuksen algoritmia tehokkaammin. Menetelmää olisi kuitenkin syytä tutkia tarkemmin esimerkiksi luomalla tilausdata jostakin jakaumasta.

Työssä lavapohjaista aktiivialuetta varten kehitetyssä mentelmässä on vielä parannettavaa. Se ei huomioi nimikkeitä, jotka kannattaisi varastoida kokonaisuudessaan aktiivialueella. Tätä varten tarvittaisiin nimikekohtaisesti lavojen yläraja tarkasteluvälillä. Se ei myöskään ota huomioon, onko nimikettä oikeasti olemassa varastolla. Algoritmia voisikin jatkokehittää juuri näiden osa-alueiden osalta. Esimerkiksi suurin osa aktiivialueesta voitaisiin jakaa nimikkeille, joilla on laskentahetkellä saldoa. Pieni osa aktiivialueesta taas voitaisiin varata nimikkeille, joilla ei laskentahetkellä ole saldoa, mutta jotka ehdottomasti kannattaa varastoida aktiivialueelle, jos nimikettä vastaanotetaan. Lavojen ylärajan voisi estimoida esimerkiksi lavojen määrällä laskentahetkellä.

Varastointisäännöillä ohjataan varastointia Motuksessa. Täyttösääntöjen ohella myös niillä on merkitystä kokonaisprosessin kannalta. Jos aktiivialueelle valituille nimikkeille määritetään myös varastointisäännöt, joilla nimike ohjautuu suoraan vastaanotosta aktiiviin, vähentyy täyttöjen määrä selvästi. Suoritetuissa simuloinneissa täytöt vähenivät n. 10-20 %. Varastointisäännöt pystytään luomaan helposti nestemallin tulosten pohjalta, mutta lavapohjaisen mallin tulosten perusteella niiden luonti on hankalaa. Varastointisäännöille olisikin hyvä lisätä maksimikappalemäärän lisäksi myös maksimilavamäärä.

Työn yhteydessä havaittiin myös simulaation selkeät hyödyt ohjausalgoritmien vertailussa. Algoritmeihin tehtyjen muutosten testaus ja vaikutus onnistuu helposti ja nopeasti simuloimalla. Simulointia kannattaisikin hyödyntää enemmän prosessien ohjausta suunniteltaessa. Niiden testaus ja kokeileminen käytännössä voi olla kallista ja aikaa vievää.

LÄHTEET

- [1] **Heikki Lahtinen and Juuso Pulli.** *Logistiikkakeskuksen kehittäjän käsikirja*. 2012. ISBN 9789529311651. Saatavissa: <http://www.eslogc.fi/fi/tuloskirja.html>.
- [2] **Jinxiang Gu.** *The forward reserve warehouse sizing and dimensioning problem*. PhD thesis, 2005. Saatavissa: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.1042&rep=rep1&type=pdf>.
- [3] **Nynke Faber, René (Marinus) B.M. De Koster, and Steef L. Van De Velde.** Linking warehouse complexity to warehouse planning and control structure: An exploratory study of the use of warehouse management information systems. *International Journal of Physical Distribution & Logistics Management*, 32(5):381–395, 2002. ISSN 0960-0035. doi: 10.1108/09600030210434161. Saatavissa: <http://www.emeraldinsight.com/10.1108/09600030210434161>.
- [4] **John J. Bartholdi and Steven T. Hackman.** *Warehouse & Distribution Science Release 0.95*. 2011. Saatavissa: <http://www.warehouse-science.com/>.
- [5] **Jinxiang Gu, Marc Goetschalckx, and Leon F. McGinnis.** Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1):1–21, February 2007. ISSN 03772217. doi: 10.1016/j.ejor.2006.02.025. Saatavissa: <http://linkinghub.elsevier.com/retrieve/pii/S0377221706001056>.
- [6] **J.P. van den Berg and W.H.M. Zijm.** Models for warehouse management: Classification and examples. *International Journal of Production Economics*, 59(1-3):519–528, March 1999. ISSN 09255273. doi: 10.1016/S0925-5273(98)00114-5. Saatavissa: <http://linkinghub.elsevier.com/retrieve/pii/S0925527398001145>.
- [7] **Tho Le-duc.** *Design and Control of Efficient Order Picking Processes*. PhD thesis, Erasmus University Rotterdam, 2005. Saatavissa: <http://repub.eur.nl/handle/6910>.
- [8] **Leanware Oy.** Leanware Logistics -varastohallintajärjestelmä. Saatavissa: <http://www.leanware.fi/logistics/varastohallintajarjestelma/>.
- [9] **B. Rouwenhorst, B. Reuter, V. Stockrahm, G.J. van Houtum, R.J. Mantel, and W.H.M. Zijm.** Warehouse design and control: Framework and literature review. *European Journal of Operational Research*, 122(3):515–533,

- May 2000. ISSN 03772217. doi: 10.1016/S0377-2217(99)00020-X. Saatavissa: <http://linkinghub.elsevier.com/retrieve/pii/S037722179900020X>.
- [10] **Charles G. Petersen and Gerald Aase.** A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, 92(1):11–19, November 2004. ISSN 09255273. doi: 10.1016/j.ijpe.2003.09.006. Saatavissa: <http://linkinghub.elsevier.com/retrieve/pii/S0925527303002937>.
- [11] **Jeroen van den Berg and A.J.R.M Gademann.** Simulation study of an automated storage/retrieval system. *International Journal of Production Research*, 38(6):1339–1356, 2000.
- [12] **Randolph W Hall.** Distance Approximations for Routing Manual Pickers in a Warehouse. *IIE Transactions*, 25(4):76–87, 1993.
- [13] **Andrei Borshchev and Alexei Filippov.** From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools. In *The 22nd International Conference of the System Dynamics Society*, 2004. Saatavissa: <http://www.econ.iastate.edu/tesfatsi/systemdyndiscreteeventabmcompared.borshchevfilippov04.pdf>.
- [14] **Tai-ran Hsu.** Application of First Order Differential Equations in Mechanical Engineering Analysis. Saatavissa: <http://www.engr.sjsu.edu/trhsu/Chapter3FirstorderDEs.pdf>.
- [15] **Eric Bonabeau.** Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99 Suppl 3:7280–7, May 2002. ISSN 0027-8424. doi: 10.1073/pnas.082080899. Saatavissa: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=128598&tool=pmcentrez&rendertype=abstract>.
- [16] **Jean Philippe Gagliardi, Jacques Renaud, and Angel Ruiz.** A simulation model to improve warehouse operations. In *2007 Winter Simulation Conference*, pages 2012–2018. Ieee, December 2007. ISBN 978-1-4244-1305-8. doi: 10.1109/WSC.2007.4419831. Saatavissa: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4419831>.
- [17] **Cornelia Triebig, Tanja Credner, Franziska Klügl, Peter Fischer, Titus Leskien, Andreas Deppisch, and Stefan Landvogt.** *Simulating Automatic High Bay Warehouses Using Agents*. 2005. ISBN 978-3-540-31731-9. doi: 10.1007/11559221_83.

- [18] **Teruaki Ito and S.M. Mousavi Jahan Abadi.** Agent-based material handling and inventory planning in warehouse. *Journal of Intelligent Manufacturing*, 13(3):201–210, 2002.
- [19] **Reino E Salmi and West Fletcher.** A Simulation Tool to Determine Warehouse Efficiencies and Storage Allocations. In *Proceedings of the 2002 Winter Simulation Conference*, pages 1274–1281, 2002.
- [20] **BK Bradley K Hollingsworth.** Decision Strategy to Minimize Replenishment Costs in a Distribution Center with Forward-reserve Storage. Master’s thesis, Ohio University, 2003. Saatavissa: <http://etd.ohiolink.edu/send-pdf.cgi/HollingsworthBradley.pdf?ohiou1056745087>.
- [21] **Rico Walter, Nils Boysen, and Armin Scholl.** The discrete forward-reserve problem – Allocating space, selecting products, and area sizing in forward order picking. 2012.
- [22] **Steven Hackman, Meir Rosenblatt, and John Olin.** Allocating Items to an Automated Storage and Retrieval System. *IIE Transactions*, 22(1):7–14, 1990.
- [23] **Matti Mononen.** Kuinka pakata matkatavarat ennen kuin aurinko sammuu eli 0/1-reppuongelman ratkaisumenetelmät teoriassa ja käytännössä. Master’s thesis, Itä-Suomen yliopisto, 2012. Saatavissa: http://epublications.uef.fi/pub/urn_nbn_fi_uef-20120146/urn_nbn_fi_uef-20120146.pdf.
- [24] **John J. Bartholdi and Steven T. Hackman.** Allocating space in a forward pick area of a distribution center for small parts. *IIE Transactions*, 40(11):1046–1053, September 2008. ISSN 0740-817X. doi: 10.1080/07408170802167662. Saatavissa: <http://www.tandfonline.com/doi/abs/10.1080/07408170802167662>.
- [25] **J Gu, M Goetschalckx, and L F McGinnis.** Solving the forward-reserve allocation problem in warehouse order picking systems. *Journal of the Operational Research Society*, 61(6):1013–1021, May 2009. ISSN 0160-5682. doi: 10.1057/jors.2009.39. Saatavissa: <http://www.palgrave-journals.com/doifinder/10.1057/jors.2009.39>.
- [26] **Søren Bloch and Christian H Christiansen.** Simultaneously Optimizing Storage Location Assignment at Forward Area and Reserve Area – a Decomposition Based Heuristic. 2008.

- [27] **J.P. van den Berg, Gunter Sharp, A.J.R.M Gademann, and Yves Pochet.** Forward-reserve allocation in a warehouse with unit-load replenishments. *European Journal of Operational Research*, (97):98–113, 1998.
- [28] **Harwin de Vries, Ruth Carrasco-gallego, Taoying Farenhorst-yuan, and Rommert Dekker.** Prioritizing Replenishments of the Forward Reserve Area. Technical report, 2012.
- [29] **Jean-Philippe Gagliardi, Angel Ruiz, and Jacques Renaud.** Space allocation and stock replenishment synchronization in a distribution center. *International Journal of Production Economics*, 115(1):19–27, September 2008. ISSN 09255273. doi: 10.1016/j.ijpe.2008.04.006. Saatavissa: <http://linkinghub.elsevier.com/retrieve/pii/S0925527308001400>.
- [30] **Jani Väihinpää.** Diplomi-insinööri, Software specialist, Leanware Oy, Tampere. Haastattelu, 16.5.2013.
- [31] **Mika Valkonen.** Diplomi-insinööri, Project Account Manager, Leanware Oy, Tampere. Haastattelu, 21.10.2013.